(72) Inventors:
• Rezzi, Francesco
  Santa Clara, California 95051 (US)
• Ozdemir, Hakan
  San Jose, California 95128 (US)

(74) Representative:
Palmer, Roger et al
PAGE, WHITE & FARRER
54 Doughty Street
London WC1N 2LS (GB)

(54) **Circuit and method for recovering synchronization information from a signal**

(57) A synchronizer circuit includes a recovery circuit coupled to the input and output terminals. The input signal includes a sync mark, and the recovery circuit is operable to recover the sync mark and generate a synchronization signal on the output terminal. Such a synchronizer circuit can recover the synchronization mark from a read signal and locate the beginning of a data stream for a separate Viterbi detector thereby reducing the complexity and increase the data-recovery speed of the Viterbi detector. The sync mark may be recovered by executing state-transition routines in alignment with the input signal.
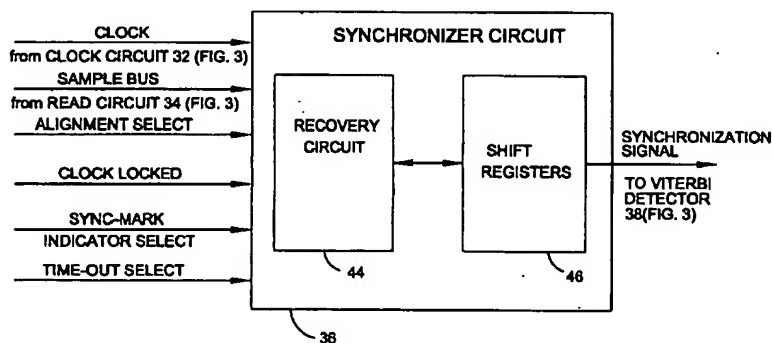
Fig.6

EP 1 089 486 A2

Description

[0001]   The invention is related generally to electronic circuits, and more particularly to a circuit and method for recovering synchronization information from a signal. In one embodiment, the circuit signals the beginning of a data stream to a Viterbi detector, and the circuit is separate from the Viterbi detector. In another embodiment, the circuit has a greater noise immunity than prior synchronization circuits, and thus can more accurately recover synchronization information from a read signal having a reduced signal-to-noise ratio (SNR). In yet another embodiment, the circuit can recover the synchronization information in fewer cycles, and thus with fewer pad bits, than prior synchronization circuits.

[0002]   Figure 1 is a partial block diagram of a conventional disk drive 10, which includes a magnetic storage disk 12 and a read channel 14 for reading data-synchronization information and encoded data from the disk 12. The read channel 14 includes a read head 16 for sensing the data-synchronization information and the encoded data stored on the disk 12 and for generating a corresponding read signal. A clock circuit 18 recovers a clock from the read signal, and a read circuit 20 amplifies the read signal, samples the read signal on the edges of the clock, and digitizes the samples. Using the data-synchronization information to locate the first data bit, the Viterbi detector 22 recovers the encoded data from the digitized samples. A decoder 24, which uses the data-synchronization information to locate the first recovered data bit from the Viterbi detector 22, decodes the recovered data.

[0003]   Figure 2 is a timing diagram of the data-synchronization information and the data stored on the disk 12 (Figure 1) in the order sensed by the read head 16 (Figure 1). The disk 12 includes a number of concentric tracks (not shown) that each include one or more respective data sectors, each sector including respective data-storage locations. Each data sector to which data has been written stores a data forerunner and the data in its storage locations. The data forerunner includes a synchronization wedge, a preamble, a synchronization mark (hereinafter sync mark), and a pad. Typically, the disk drive 10 (Figure 1) writes a respective wedge at the beginning of each data sector during the formatting of the disk 12, and writes the preamble, sync mark, and pad to a data sector each time one writes data to the data sector. As the disk 12 rotates, the read head 16 first senses the wedge at time t_0, and then senses the preamble, sync mark, pad, and data at relative times t_1, t_2, t_3, and t_4, respectively.

[0004]   Referring to Figures 1 and 2, the read channel 14 operates as follows. A front-end circuit (not shown) receives the read signal and activates the clock circuit 18 in response to the synchronization wedge. Once activated, the clock circuit 18, which typically includes a phase-locked loop (PLL, not shown), aligns the phase and frequency of the clock signal with the phase and frequency of the preamble. Next, the Viterbi detector 22 recovers from the sync mark the time — typically the clock edge — at which the detector 22 will receive the first data sample. The pad includes a number of don't-care bits, and thus provides a delay between the end of the sync mark and the beginning of the data. This delay allows the detector 22 to reliably recover this first-data-sample time before it actually occurs. The detector 22 then begins recovering the data from the read signal at the first-data-sample time. After a delay equal to its latency, the detector 22 provides the first recovered data bit to the decoder 24 at a first-recovered-bit time, and synchronizes the decoder 24 such that it begins decoding the recovered data at the first-recovered-bit time. But as discussed below, if the detector 22 fails to accurately recover the first-data-sample time, then it begins recovering the data at the wrong sample time, and thus typically generates fatal read errors.

[0005]   One problem with the Viterbi detector 22 is that it often requires the read signal to have a relatively high signal-to-noise ratio (SNR), and thus often limits the data-storage density, and thus the data-storage capacity, of the disk 12.

[0006]   The storage density of the disk 12 is a function of the distances between the storage locations within the data sectors and the distances between the disk tracks. The smaller these distances, the greater the storage density, and vice-versa. The storage capacity of the disk 12 is proportional to its surface area and its storage density. But because the diameter of the disk 12, and thus its surface area, is typically constrained to industry-standard sizes, the option of increasing the surface area of the disk 12 to increase its storage capacity is usually unavailable to disk-drive manufacturers. Therefore, increasing the storage density is typically the only available technique for increasing the storage capacity of the disk 12.

[0007]   Typically, the greater the storage density of the disk 12, the closer the surrounding storage locations are to the read head 16 while it is reading the surrounded storage location, and thus the lower the signal-to-noise ratio (SNR) of the read signal. Specifically, the closer the surrounding locations are to the read head 16, the greater the magnitudes of the magnetic fields that these locations respectively generate at the head 16, and thus the greater the Inter Symbol Interference (ISI). The greater the ISI, the smaller the root-mean-square (RMS) amplitude of the read signal. In addition, as the storage density of the disk 12 increases, the media noise also increases. Generally, the media noise results from the uncertainty in the shapes of the read pulses that compose the read signal. This uncertainty is caused by unpredictable variations in the relative positions of the storage locations from one data-write cycle to the next. Moreover, for a given spin rate of the disk 12, as one increases the linear storage density within the data sectors, he/she must also increase the bandwidth of the read head 16 to accommodate the increased number of storage locations that the read head 16 must sense in a given time period. This increase in bandwidth causes a proportional increase in the white

noise generated by the read head 16. The SNR of the read signal for a particular storage location is the ratio of the RMS amplitude of the corresponding portion of the read signal to the sum of the amplitudes of the corresponding media and white noise. Thus, the lower the RMS amplitude of the read signal and the greater the amplitudes of the media and/or white noise, the lower the SNR of the read signal.

5 [0008] Unfortunately, as the SNR of the read signal decreases, the data-recovery speed of the Viterbi detector 20 often decreases as well. Specifically, the lower the SNR of the read signal, the lower the accuracy of the detector 20. As discussed above, the failure of the detector 20 to accurately recover the first-data-sample time from sync mark often causes serious read errors. If the error processing circuit (not shown) initially detects a read error, then it tries to correct the error using conventional error-correction techniques, If the processing circuit cannot correct the error using these

10 techniques — typically the case when the detector 20 recovers an inaccurate first-data-sample time — then it identifies the error as "fatal" and instructs the read channel 14 to re-read the data from the disk 12. The time needed by the processing circuit for error detection and error correction and the time needed by the read channel 14 for data re-read increase as the number and severity of the read errors increase. As the error-processing and data re-read times increase, the effective data-read speed of the channel 14, and thus of the disk drive 10, decreases.

15 [0009] Therefore, to maintain an acceptable effective data-read speed, the manufacture rates the Viterbi detector 22 for a minimum read-signal SNR. Unfortunately, if the SNR of the read signal falls below this minimum, then the accuracy of the read channel 14 often degrades such that at best, the effective data-read speed of the disk drive 10 falls below its maximum rated speed, and at worst, the disk drive 10 cannot accurately read the stored data.

[0010] Referring again to Figures 1 and 2, another problem is that the Viterbi detector 22 recovers both the data-

20 synchronization information and the data. Unfortunately, this dual functionality often increases the circuit complexity and limits the effective data-recovery speed of the detector 22.

[0011] Furthermore, including the pad in the data forerunner reduces the amount of data that the respective data sector can hold. But eliminating or reducing the length of the pad may decrease the sync-recovery accuracy of the Viterbi detector 22, and thus may increase the probability of a fatal read error that requires the read channel 14 to

25 reread the data.

[0012] Detailed descriptions of the structure and operation of a conventional Viterbi detector such as the Viterbi detector 22 are available in many references and in the background section of heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-176 (1678-6)).

[0013] In one aspect of the invention, a synchronizer circuit includes an input terminal, an output terminal, and a

30 recovery circuit coupled to the input and output terminals. The input terminal receives an input signal that includes a sync mark, and the recovery circuit is operable to recover the sync mark from the input signal and to generate a synchronization signal on the output terminal in response to the recovered synchronization mark.

[0014] For example, such a synchronizer circuit can recover the synchronization mark from a read signal and locate the beginning of a data stream for a Viterbi detector that is separate from the circuit. By performing the sync-recovery

35 function in a separate circuit, one can reduce the complexity and increase the data-recovery speed of the Viterbi detector.

[0015] In another aspect of the invention, the synchronizer circuit recovers the sync mark by executing state-transition routines in alignment with the input signal.

[0016] For example, one can align the synchronizer circuit's state-transition routines to the preamble of the read

40 signal. Such alignment increases the circuit's noise immunity, and thus allows the circuit to recover the sync mark from a read signal having a SNR that is lower than the minimum read-signal SNR of prior sync-recovery circuits. Furthermore, such alignment reduces the time needed for the circuit to reliably detect the sync mark, and thus allows one to shorten the pad of the data forerunner.

45 Figure 1 is a partial block diagram of a disk-drive that includes a storage disk and a read channel according to the prior art.

Figure 2 is a timing diagram of the data forerunner and the data in the order read from the disk of Figure 1 according to the prior art.

Figure 3 is a partial block diagram of a read channel according to an embodiment of the invention.

50 Figure 4 is a timing diagram of the clock and read signals of Figure 3 during the reading of the preamble according to an embodiment of the invention.

Figure 5 is a timing diagram of the read, clock, and sync signals of Figure 3 during the reading of the data according to an embodiment of the invention.

Figure 6 is a block diagram of the sync circuit of Figure 3 according to an embodiment of the invention.

55 Figure 7 is a diagram of a preamble, sync mark, pad, and sync indicator according to an embodiment of the invention.

Figure 8 is a trellis diagram for the sync circuit of Figure 6 according to an embodiment of the invention.

Figure 9 is a diagram of the possible sequences that the recovery circuit of Figure 6 can traverse through the trellis

diagram of Figure 8 during the reading of the preamble of Figure 7.

Figure 10A is the trellis diagram of Figure 8 at a sample time during the generation of the sync indicator by the sync circuit of Figure 4 according to an embodiment of the invention.

Figure 10B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 10A.

Figure 11A is the trellis diagram of Figure 10A at a subsequent sample time.

Figure 11B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 11A.

Figure 12A is the trellis diagram of Figure 11A at a subsequent sample time.

Figure 12B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 12A.

Figure 13A is the trellis diagram of Figure 12A at a subsequent sample time.

Figure 13B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 13A.

Figure 14A is the trellis diagram of Figure 13A at a subsequent sample time.

Figure 14B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 14A.

Figure 15A is the trellis diagram of Figure 14A at a subsequent sample time.

Figure 15B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 15A.

Figure 16A is the trellis diagram of Figure 15A at a subsequent sample time.

Figure 16B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 16A.

Figure 17A is the trellis diagram of Figure 16A at a subsequent sample time.

Figure 17B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 17A.

Figure 18A is the trellis diagram of Figure 17A at a subsequent sample time.

Figure 18B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 18A.

Figure 19A is the trellis diagram of Figure 18A at a subsequent sample time.

Figure 19B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 19A.

Figure 20A is the trellis diagram of Figure 19A at a subsequent sample time.

Figure 20B shows the contents of the sync-circuit shift registers of Figure 4 corresponding to the trellis diagram of Figure 20A.

Figure 21 is the trellis diagram of Figure 20A showing the surviving paths only.

Figure 22 is a block diagram of a disk-drive system that incorporates the sync circuit of Figure 4 according to an embodiment of the invention.

[0017]   Figure 3 is a partial block diagram of a read channel 30 according to an embodiment of the invention. The channel 30 includes a clock circuit 32, a read circuit 34, a synchronization circuit 36 for recovering synchronization information from a read signal, an alignment circuit 37 for aligning the circuit 36 to the read signal, and a Viterbi detector 38 for recovering data from the read signal. The channel 30 may also include a decoder 39 for decoding the recovered data. Because the Viterbi detector 38 need not recover the synchronization information, it can have simpler circuitry and recover data faster than prior Viterbi detectors. In one embodiment, the channel 30 is compatible with an EPR4 data-recovery protocol, the Viterbi detector 38 is similar to the Viterbi detector described in heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-176 (1678-6)), and the decoder 39 is similar to the decoder described in heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-167 (1678-5)). In another embodiment discussed below, the alignment circuit 37 causes the sync circuit 36 to recover the sync information by executing its state-transition routines in alignment with the read signal. This alignment increases the noise immunity of the circuit 36, and thus allows the circuit 36 to recover the sync information from a read signal having a SNR that is lower than the minimum read-signal SNR specified for prior Viterbi detectors. Furthermore, such alignment reduces the time needed for the circuit to reliably detect the sync information, and thus allows one to reduce the length of the pad within the data forerunner.

[0018]   Referring to Figures 3 -5, the operation of the read channel 30 is described according to an embodiment of the invention.

[0019]   Figure 4 is a timing diagram of the read, clock, and clock-locked signals of Figure 3 as the read head (not shown in Figure 3) reads the preamble from a data sector (not shown). In one embodiment, the preamble stored in the

data sector is 0011001100110011 . . . and is no more than 200 bits long.

[0020] Ignoring noise for purposes of illustration, the read signal is a sinusoid as the read head reads the preamble. In terms of the EPR4 samples B — the details of the EPR4 protocol are discussed in heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-176 (1678-6)) — the read signal has a positive peak amplitude of +2 and a negative peak amplitude of -2. Referring to Figures 3 and 4, the clock circuit 32 locks the clock signal to the read signal while the read head reads the preamble. A conventional front-end circuit (not shown) enables the clock circuit 32 to begin generating the clock signal in response to the first zero crossing of the read signal at time t_0. Specifically, at t_0, the clock circuit 32 generates a falling edge of the clock signal, and, over the next few cycles of the read signal, aligns the subsequent falling clock edges with the respective zero crossings of the read signal as shown. Therefore, the clock signal is twice the frequency of the read signal as the read head reads the preamble.

[0021] After it has locked the phase and frequency of the clock signal to the read signal, the clock circuit 32 transitions the clock-locked signal from one logic level to the other — from logic 0 to logic 1 in this embodiment — at time t_1 to indicate that the clock signal is locked. In one embodiment, the manufacturer empirically determines the lock time, and programs the clock circuit 32 to transition the clock-locked signal after the elapse of this predetermined lock time as measured from t_0.

[0022] The read circuit 34 uses the clock signal to sample the read signal. Specifically, a first analog-to-digital (A/D) converter 40 of the read circuit 34 generates respective samples A/D-0 of the read signal on the rising clock edges, and a second A/D converter 42 generates respective samples A/D-1 of the read signal on the falling clock edges. A finite-impulse-response (FIR) filter 43 equalizes the samples A/D-0 and A/D-1 in a conventional manner to generate respective equalized read-signal samples B0 and B1 on the rising and falling clock edges, respectively. The sync circuit 36 simultaneously processes two respective samples B0 and B1 on each rising clock edge.

[0023] Still referring to Figures 3 and 4 and, as discussed below in conjunction with Figure 9, in one embodiment, if the state transitions of the sync circuit 36 are improperly aligned to the read signal as the read head reads the preamble, then the alignment circuit 37 forces the circuit 36 into proper alignment. One can program the circuit 36 to begin operating in response to the first rising clock edge after t_0, or in response to the first rising clock edge after the clock-locked time t_1. Once the circuit 36 begins operating, the alignment circuit 37 determines the alignment of the circuit 36 state transitions with respect to the read signal. If the circuit 37 determines that this alignment is proper, then it does nothing. Conversely, if the circuit 37 determines that this alignment is improper, then it delays the execution of the circuit 36 state transitions until the circuit 36 is in proper alignment with the read signal.

[0024] Figure 5 is a timing diagram of the read, clock, and synchronization signals of Figure 3 during reading of the data according to an embodiment of the invention. After the read head (not shown in Figure 3) finishes reading the preamble, it reads the sync mark, which identifies the rising clock edge at time t_2. Time t_2 corresponds to the read head reading and the read circuit 36 sampling the first bit of data in the data sector. In response to this rising clock edge at t_2, the A/D converter 40 generates a sample A/D-0 that corresponds to the first data bit. Likewise, in response to the subsequent falling clock edge at t_3, the A/D converter 42 generates a sample A/D-1 that corresponds to the second data bit. Next, in response to the rising and falling clock edges t_4 and t_5, the FIR filter 43 generates samples B0 and B1 that respectively correspond to the first and second data bits. Consequently, to insure that these first two bits of data are recovered, the Viterbi detector 38 begins its data-recovery routine on the subsequent rising clock edge at t_6. Therefore, as discussed below in conjunction with Figures 10A - 21, the sync circuit 36 recovers the first-data-sample time t_2 from the sync mark and transitions the Viterbi sync signal from one logic level to the other — from logic 0 to logic 1 in this embodiment — in response to the next rising clock edge at t_4. In response to this transition of the Viterbi sync signal, the Viterbi detector 38 begins recovering the data on the next rising clock edge at t_6. The detector 38 has a latency, which is a delay from the time t_6 to a time t_7 at which the detector 38 provides the recovered first and second data bits at its output. In one embodiment, the latency is twenty four clock cycles, *i.e.*, forty eight bits. Therefore, in response to the rising clock edge at t_7, the sync circuit 36 transitions the detector sync signal from one logic level to the other — from logic 0 to logic 1 in this embodiment. In one embodiment, the sync circuit 36 is programmed with the latency of the detector 38 such that it transitions the Viterbi sync signal, waits the number of clock cycles equal to the latency, and then transitions the detector sync signal if and only if it has previously transitioned the Viterbi sync signal. In response to this transition of the detector sync signal, the decoder 39 begins decoding the recovered data at the next rising clock edge at time t_8.

[0025] Still referring to Figures 3 and 5, although the FIR filter 43 is described as introducing a delay of one clock cycle between the generation of A/D-0, A/D-1 and B0, B1 respectively, it may introduce a different delay. For example, in one embodiment, the filter 43 introduces a delay of three clock cycles. But regardless of the length of this delay, the sync circuit 36 transitions the Viterbi sync signal in response to the rising clock edge (t_4) at which the filter 43 generates the sample B0 corresponding to the first data bit, and transitions the detector sync signal in response to the rising clock edge (t_7) at which the Viterbi detector 38 generates the first recovered data bit.

[0026] Figure 6 is a block diagram of the synchronizer circuit 36 of Figure 3 according to an embodiment of the invention. The circuit 36 includes a recovery circuit 44 for tracking the preamble and sync mark of the read signal, for

recovering the first-data-sample time t_2 (Figure 5) from the sync mark, and for generating a sync indicator. The circuit 36 also includes shift registers 46 for storing the sync indicator. The circuit 36 transitions the Viterbi sync signal in response to the stored sync indicator as discussed below in conjunction with Figures 10A - 20B. In addition to receiving the clock and clock-locked signals from the clock circuit 32, the sample B0 and B1 from the read circuit 34 via the sam-
5   ple bus, and the alignment signal from the alignment circuit 37, the circuit 36 receives sync-indicator-select and time-out-select signals. As discussed below in conjunction with Figures 10A - 21, the sync-indicator-select signal is a logic signal that, depending on its state, causes the circuit 36 to generate the Viterbi sync signal in response to a partially generated sync indicator or in response to only a fully generated sync indicator. And as discussed below in conjunction with Figures 10A - 21, the time-out-select signal is a logic signal having an active state that causes the circuit 36 to halt
10   generation of the Viterbi and detector sync signals if the recovery circuit 44 fails to generate the sync indicator within a predetermined time period.

[0027]   Referring to Figures 7 - 21, the operation of the synchronizer circuit 36 of Figure 6 is discussed according to an embodiment of the invention.

[0028]   Figure 7 is a diagram of a preamble, sync mark, and pad according to an embodiment of the invention, and
15   sync indicator generated by the synchronizer circuit 36 (Figure 6) according to an embodiment of the invention. The pre-amble has the same bit pattern, 00110011. . . , as discussed above in conjunction with Figure 4. The sync mark 50 includes four groups 52, 54, 56, and 58 of four bits that are inverted with respect to the preamble. The number of non-inverted separator bits between adjacent groups decreases as the sync mark progresses. Specifically, there are six separator bits between the groups 52 and 54, four separator bits between the groups 54 and 56, and two separator bits
20   between the groups 56 and 58. These different separations allow the circuit 36 to identify the beginning and end of the sync mark. One can highlight the groups 52, 54, 56, and 58 by XORing the preamble pattern and the sync mark 50 (PR ⊕ SM). PR ⊕ SM includes all logic 0's except for four groups 60, 62, 64, and 66 of logic 1's, which respectively corre-spond to the inverted groups 52, 54, 56, and 58 of the sync mark 50. The sync indicator includes four indicator bits 68, 70, 72, and 74 — logic 1's in this embodiment — which respectively correspond to the last bits in the groups 52, 54, 56,
25   and 58 of the sync mark. Consequently, nine bits — logic 0's in this embodiment — separate the indicator bits 68 and 70, seven bits separate the indicator bits 70 and 72, and five bits separate the indicator bits 72 and 74.

[0029]   Referring to Figures 6 and 7, once the synchronizer circuit 36 has recovered the sync mark and has thus generated the sync indicator, it generates the Viterbi and detector sync signals as discussed above in conjunction with Figure 5. Specifically, the circuit 36 is programmed to generate the Viterbi sync signal on the first rising clock edge after
30   the read head (not shown in Figures 6 or 7) reads the pad 76, which is six bits long in one embodiment. Because the circuit 36 processes two samples B0 and B1 per clock cycle (Figure 3), and because each sample corresponds to a respective bit of the data forerunner (Figure 2), there are three rising clock edges within the preamble 76. Therefore, the circuit 36 transitions the Viterbi sync signal in response to the third rising clock edge after generating the last indi-cator bit 74 of the sync indicator. The circuit 36 then waits the latency period of the Viterbi detector 38 before transition-
35   ing the detector sync signal. As discussed below in conjunction with Figures 10A -21, although all the surviving paths generated by the recovery circuit 44 may not have merged at the indicator bit 74 by this third rising clock edge, typically a sufficient number of the paths have merged to warrant accurately timed transitions of the Viterbi and detector sync signals. Alternatively, the pad 76 can be lengthened to equal or exceed the latency of the recovery circuit 44 so that all the paths have a chance to merge at the indicator bit 74 before the circuit 36 transitions the Viterbi sync signal. Although
40   this latter procedure is more reliable, the former procedure allows the pad 76 to have fewer pad bits than many prior pads, and thus allows the respective data sector to hold more data bits.

[0030]   Still referring to Figures 6 and 7 and as discussed above in conjunction with Figure 6, one embodiment of the synchronizer circuit 36 has a partial-sync-mark-recovery mode that, when enabled, causes the circuit 36 to transi-tion the Viterbi and detector sync signals even if the recovery circuit 44 has recovered only part of the sync mark, and
45   thus has generated only part of the sync indicator. Specifically, if the sync-indicator-select signal has an inactive state, then the partial-sync-mark-recovery mode is disabled such that the circuit 36 transitions the Viterbi and detector sync signals only if the recovery circuit 44 generates the entire sync indicator as discussed in the preceding paragraph. Con-versely, if the sync-indicator-select signal has an active state, then the partial-sync-mark-recovery mode is enabled such that the circuit 36 transitions the Viterbi and detector sync signals if the circuit 44 generates at least three of the
50   indicator bits 68, 70, 72, and 74 and the separator bits in between. The inventors have found that generating the three of the four indicator bits and the separator bits in between still yields a reasonably accurate recovery of the sync mark, and thus yields reasonably accurate transitions of the Viterbi and detector sync signals.

[0031]   For example, if the circuit generates the indicator bits 68, 70, and 72 and the separator bits in between and the partial-sync-mark-recovery mode is enabled, then the circuit 36 transitions the Viterbi sync signal on the third rising
55   clock edge during the reading of the pad 76. Thus, the failure of the recovery circuit 44 to generate one of the indicator bits does not prevent the circuit 36 from transitioning the Viterbi or detector sync signals. One reason that the circuit 44 may fail to generate an indicator bit is a read-signal SNR that is too low during the reading of the sync mark.

[0032]   Also as discussed above in conjunction with Figure 6, one embodiment of the synchronization circuit 36 has

a time-out mode that, when enabled, causes the circuit 36 to stop searching for the sync mark and to disable the transitioning of the Viterbi and detector sync signals after a predetermined time has elapsed. This prevents erroneous detection of the sync mark while the read head is reading the data. Specifically, if the time-out-select signal has an inactive state, then the time-out mode is disabled such that the circuit 36 searches for the sync mark until the recovery circuit 44 recovers the mark or until a control circuit (not shown in Figure 6) disables the circuit 36. If, however, the circuit 44 fails to recover the sync mark from the data forerunner as discussed in the preceding two paragraphs, it may recover a false sync mark within the data, particularly if the data includes a pattern that is similar or identical to the sync mark. Conversely, if the time-out-select signal has an active state, then the time-out mode is enabled such that after the recovery circuit 44 generates all but one of the required indicator bits 68, 70, 72, and 74, the circuit 36 disables the circuit 44 and does not transition the Viterbi or detector sync signals unless the circuit 44 generates a remaining indicator bit within a predetermined time period. In one embodiment, this time period is equivalent to thirty bits, i.e., fifteen clock cycles, measured from the generation of the penultimate indicator bit. For example, suppose that the time-out mode is enabled such that the circuit 36 transitions the Viterbi and detector sync signals if the circuit 44 generates any three of the indicator bits 68, 70, 72, and 74. Furthermore, suppose that the recovery circuit 44 recovers the inverted groups 52 and 54 of the synchronization mark and thus generates the indicator bits 68 and 70. If the circuit 44 does not recover at least one of the groups 56 and 58, and thus does not generate at least one of the indicator bits 72 and 74, within fifteen clock cycles, then it stops searching for the sync mark, and the circuit 36 does not transition the Viterbi or detector sync signals. After a predetermined system time out, a disk-drive control circuit (not shown in Figure 6) instructs the read channel 30 (Figure 3) to reread the data sector. This saves time, because the control circuit starts the reread sooner than if the circuit 44 recovered a false sync mark, the sync circuit 36 transitioned the Viterbi and detector sync signals at inaccurate times, the Viterbi detector 38 (Figure 3) generated a fatal read error, and the control circuit performed the error-detection/correction procedures before it detected the fatal error and forced a sector reread.

[0033]   Still referring to Figure 7, the preamble, sync mark 50, pad 76, and the sync indicator may have different patterns than those illustrated. For example, the mark 50 may have more or fewer than twenty eight bits and the groups 52, 54, 56, and 58 may be separated by different numbers of bits. In addition, the pad may have more or fewer than six bits.

[0034]   Figure 8 is a time-varying trellis diagram 80, which represents a sync-mark-recovery/sync-indicator-generation algorithm that the recovery circuit 44 of Figure 6 executes according to an embodiment of the invention. This algorithm is compatible with an EPR4 data-recovery protocol, and is similar to recovery algorithms implemented by conventional Viterbi detectors. Heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-176 (1678-6)) includes a detailed description of the EPR4 data-recovery protocol and the operation of EPR4-compatible Viterbi detectors. For brevity, this description is not repeated here.

[0035]   Still referring to the trellis 80 of Figure 8, at each relative sample time k - k+1, the three most recent bits of the binary data forerunner have one of four possible states S: S0 = 000, S1 = 001, S6 = 110, and S7 = 111. Because the trellis 80 is designed for the preamble, sync mark, and pad of Figure 7 and because the state transitions of the circuit 44 (Figure 6) are aligned with the read signal as discussed below in conjunction with Figure 9, the data forerunner does not have the states S2 = 101, S3 = 011, S4 = 100, and S5 = 101. Therefore, the states S2 - S5 are omitted from, the trellis 80. This significantly simplifies the trellis 80 and the recovery circuit 44. The trellis 80 includes one column of state circles 82 for each respective sample time k - k+1. Within each circle 82, the right-most bit 84 represents a possible value for the most recent bit of the data forerunner at the respective sample time, the middle bit 86 represents a possible value for the second most recent bit of the data forerunner, and the left-most bit 88 represents a possible value for the third most recent bit of the data forerunner. Because the recovery circuit 44 process two samples B0 and B1 at each sample time k and k+1, the bits 84 and 86 within each circle 82 are the most recent bits corresponding to the respective sample time k and k+1. For example, in the circle 82b, the bits 84b, 86b, and 88b represent the possible values — logic 1, logic 0, and logic 0, respectively — for the three most recent bits of the data forerunner at sample time k. The recovery circuit 44 processes the samples B0 and B1 respectively corresponding to the bits 84b and 86b during the sample time k, and processes the sample B1 corresponding to the bit 88b during the previous to the sample time.

[0036]   Furthermore, a finite number of potential state transitions exist between the states S at one sample time k - k+1 of the trellis 80 and the states S at the next respective sample time. "Branches" 90 represent these possible state transitions. For example, starting at the possible state S0 (circle 82a) at sample time k, the only choice for the next state S at k+1 is S1 (circle 82f). Thus, the branch 90a represents this possible state transition. Because the patterns of the preamble, sync mark, and pad are known, some of the possible branches are omitted from the trellis 80. This omission further simplifies the trellis 80 and the recovery circuit 44 (Figure 6). In addition, the branches 90a and 90l, which are in bold line, represent the two state transitions for which the recovery circuit 44 generates a potential indicator bit 68, 70, 72, or 74 (Figure 7) as discussed below in conjunction with Figures 10A - 21.

[0037]   The trellis 80 also includes respective state-transition labels for clarity. Specifically, the values 92 and 94 are the respective values of the next two data-forerunner bits 84 and 86 that the respective branch 90 points to, and the values 96 and 98 are the respective values of the corresponding samples B1 and B0. For example, for the state transition

represented by the branch 90a, the values 92a and 94a respectively represent the bits 84f and 86f, and thus respectively equal logic 1 and logic 0, and the corresponding sample values 96a (B1) and 96b (B0) respectively equal +1 and 0, respectively.

[0038]   Still referring to Figure 8, as the recovery circuit 44 (Figure 6) traverses the trellis 80, it calculates the branch metrics and updates the path metrics in a conventional manner such as that described in U.S. Patent 5,430,744, "Method and Means for Detecting Partial Response Waveforms Using A Modified Dynamic Programming Heuristic," which is incorporated by reference.

[0039]   Referring to Figures 3, 4, 8, and 9, in one embodiment, the synchronizer circuit 36 aligns the state transitions executed by the recovery circuit 44 (Figure 6) and represented by the trellis 80 with the data forerunner to promote accurate recovery of the sync mark and accurate generation of the sync indicator.

[0040]   Figure 9 is a diagram of the possible state-transition sequences A - D that the recovery circuit 44 (Figure 6) can traverse through the trellis 80 while the read head reads the preamble of Figure 7. Referring to Figure 4, the traversed sequence depends on the value of the read signal, and thus the value of the preamble, at the first rising clock edge after time t_0 or t_1, depending on whether the synchronizer circuit 36 is programmed to begin operation before or after the transition of the clock-locked signal. For example purposes the former case is discussed, it being understood that operation of the circuit 36 in the latter case is similar.

[0041]   Referring to Figure 8, the first rising clock edge after t_0 causes the recovery circuit 44 to execute a transition from the states S at sample time k to the states S at k+1. Likewise, the second rising clock edge after t_0 causes the recovery circuit 44 to execute a transition from the states S at k+1 to the states S at k. These first and second state transitions are "junk" transitions because the sync circuit 36 processes inaccurate samples B0 and B1 during the first transition and an inaccurate sample B0 during the second transition. This is due to the pipe-line delay of the read circuit 34. Assuming that the alignment between the clock signal and the read signal is as shown in Figure 4 and assuming a noiseless read signal, the first rising clock edge causes the A/D 40 to generate A/D-0 = -2 , and the following falling clock edge causes the A/D 42 to generate A/D-1 = 0 . Similarly, the second rising clock edge causes the FIR filter 43 to generate B0 = -2, and the following falling clock edge causes the filter 43 to generate B1 = 0. Therefore, on the third rising clock edge after t_0, the recovery circuit 44 executes a transition from the states S at k+1 to the states S at k with B0 = -2 and B1 = 0. According to the trellis 80, these values for B0 and B1 indicate that the preamble has the state S1 at sample time k. Thus, in this example, the recovery circuit 44 traverses the trellis 80 via the sequence A.

[0042]   In one embodiment, the synchronizer circuit 36 aligns the state transitions of the recovery circuit 44 with the preamble such that the recovery circuit 44 traverses the trellis 80 (Figure 8) via the sequence A. As discussed below in conjunction with Figures 10A - 21, this alignment insures that the recovery circuit 44 accurately recovers the sync mark and generates the sync indicator (Figure 7).

[0043]   Referring to Figures 4 and 9, because the clock circuit 32 (Figure 3) generates the first falling clock edge at time t_0 in response to a zero crossing of the read signal, then, absent some peculiar glitch, the recovery circuit 44 traverses the trellis 80 via the sequence A or the sequence D. For example, as discussed above, if the clock signal is aligned with the read signal as shown in Figure 4, then the circuit 44 traverses the trellis 80 via the sequence A. In this alignment, the first falling clock edge is aligned with a negative-slope zero crossing of the read signal between +2 and -2. Conversely, if the first falling clock edge is shifted 180° with respect to the read signal and thus is aligned with a positive-slope zero crossing between -2 and +2, then the circuit 44 traverses the trellis 80 via the sequence D.

[0044]   Referring to Figures 3 and 9, the alignment circuit 37 determines the alignment of the state transitions of the recovery circuit 44 with respect to the preamble and, if necessary, realigns these state transitions so that the circuit 44 traverses the trellis 80 via the sequence A.

[0045]   To determine the alignment of the circuit 44 state transitions, the circuit 37 implements the following equation:

$$\text{Count} = \text{Count} + \text{sign(B0)} \times (-1)^{CC} \tag{1}$$

where Count is initialized to equal 0, CC is the number of the clock cycle, and sign(B0) is the polarity of the sample B0 taken at the rising clock edge that begins the clock cycle CC. If the circuit 44 is traversing the trellis 80 via the sequence A, then, after a number of clock cycles CC, Count will equal a positive value. Conversely, if the circuit 44 is traversing the trellis 80 via the sequence D, then, after a number of clock cycles CC, Count will equal a negative value. For example, referring to Figure 4, the first rising clock edge after time t_0 represents the beginning of the clock cycle corresponding to CC = 1. On the third rising clock edge after t_0 corresponding to CC = 3, the alignment circuit 37 processes B0 = -2 from the FIR filter 43, and according to equation (1), Count = 0 + -(-1)^3 = +1. (As discussed above in conjunction with Figure 5, the first valid sample B0= -2 is unavailable to the circuit 37 until the third rising clock edge because of the pipe-line delays introduced by the A/D converters 40 and 42 and the FIR filter 43.) On the fourth rising clock edge, CC = 4, B0 = +2, and Count = 1 + (-1)^4 = +2. On the fifth rising clock edge, CC = 5, B0 = -2, and Count = 2 + -(-1)^5 = +3. Therefore, Count is increasing and eventually becomes a positive number — despite any junk values calculated for Count

8

during clock cycles 1 and 2 — when the circuit 44 traverses the trellis 80 via the sequence A. Conversely, if the clock signal of Figure 4 is shifted 180° with respect to the read signal, then the circuit 44 traverses the trellis 80 via the sequence D and Count is decreasing and eventually becomes a negative number. Therefore, after a predetermined number of clock cycles, the alignment circuit 37 analyzes the polarity of Count. If Count equals a positive number, then the circuit 37 determines that the state transitions of the recovery circuit 44 are properly aligned with the preamble and thus does not realign the circuit 44. Conversely, if Count equals a negative number, then the circuit 37 determines that the state-transitions of the recovery circuit 44 are improperly aligned with the preamble and thus realigns the circuit 44 as discussed below.

[0046]   The above-described analysis of equation (1) is valid as long as CC has the above-disclosed relationship with the sample clock. Specifically, the analysis is valid as long as CC equals an odd number for the clock cycle during which the alignment circuit 37 processes the first valid sample B0 of the preamble. CC has this characteristic as long as the combined pipeline delay introduced by the A/D converters 40 and 42 and the FIR filter 43 is an even number of clock cycles. For example, in the above-described analysis of equation (1), this combined pipeline delay is two clock cycles.

[0047]   If CC has a different relationship with the sample clock, then the alignment circuit 37 can implement a modified version of equation (1) or a modified analysis thereof to obtain accurate alignment information. For example, suppose the pipeline delay introduced by the A/D converters 40 and 42 and the FIR filter 43 is an odd number of clock cycles. Consequently, CC equals an even number for the clock cycle during which the alignment circuit 37 processes the first valid sample B0 of the preamble. This causes Count to equal a negative value if the state transitions of the recovery circuit 44 are properly aligned with the preamble, and causes Count to equal a positive value if the state transitions are improperly aligned. Therefore, one can modify the circuit 37 to interpret a negative Count value as proper alignment and to interpret a positive Count value as improper alignment.

[0048]   In one embodiment as discussed above, the synchronizer circuit 36 begins operating on the first rising clock edge after t_0 (Figure 4), and thus the circuit 37 is programmed to begin implementing equation (1) starting with this first rising clock and to analyze Count after sixteen clock cycles. Sixteen clock cycles are typically sufficient to allow the clock circuit 32 to properly lock the clock signal to the read signal and to allow the circuit 37 to analyze a sufficient number of samples B0 for an accurate indication of the alignment between the circuit 44 and the preamble.

[0049]   In another embodiment as discussed above, the circuit 37 is programmed to begin implementing equation (1) starting with the first rising clock edge after t_1 (Figure 4) and to analyze Count after eight clock cycles. Because the clock signal is locked before the circuit 37 implements equation (1), fewer clock cycles are needed — eight instead of sixteen in this embodiment — for an accurate indication of the alignment between the circuit 44 and the preamble.

[0050]   Still referring to Figures 3 and 9, if the alignment circuit 37 determines that the circuit 44 is traversing the trellis 80 via the undesired sequence D, then the circuit 37 realigns the circuit 44 by causing the circuit 44 to skip a state transition. Referring to Figure 9, the sequences A and D are the same except that they are offset by one sample time. Specifically, the sequence A has the state S1 at sample time k of the trellis 80 and has the state S6 at sample time k+1. Conversely, the sequence D has the state S6 at sample time k and has the state S1 at sample time k+1. Therefore, the circuit 44 can shift between states A and D if it "waits" one clock cycle. For example, suppose the circuit 44 is traversing the trellis 80 via the sequence D and the circuit 44 has just executed a state transition between k and k+1. Therefore, the preamble has the state S1 at k+1. Next, the alignment circuit 37 conventionally prohibits the circuit 44 from executing a state transition between k+1 and k at the next rising clock edge. Therefore, the preamble transitions to the state S6 at this next rising clock edge while the circuit 44 remains at sample time k+1. This is the alignment for the sequence A. Then, the circuit 37 enables the circuit 44 to transition between k+1 and k at the following rising clock edge such that the circuit 44 begins traversing the trellis 80 via the desired sequence A.

[0051]   Referring to Figures 10A - 21, after the state transitions of the recovery circuit 44 are aligned with the preamble, the circuit 44 recovers the sync mark and generates the sync indicator according to an embodiment of the invention. Figures 10A, 11A, . . ., and 21 show an expanded trellis diagram 80 (Figure 8) and the surviving paths at respective sample times t - t+11, and Figures 10B, 11B, . . ., and 20B show the contents of four corresponding shift registers Reg0, Reg1, Reg6, and Reg7 — one for each state S0, S1, S6, and S7 — at these respective sample times. These registers compose the block of shift registers 46 of Figure 6. The state-transition branches 90 from the even sample times t, t+2, . . ., t+10 to the respective odd sample times t+1, t+3, . . ., t+11 are the same as the state-transition branches 90 from k to k+1 of Figure 8. Likewise, the state-transition branches from the odd sample times to the respective even sample times are the same as the state-transition branches 90 from k+1 to k of Figure 8. For clarity, the sample-time labels "k" and "k+1" show the relationship between the expanded trellis 80 of Figures 10A, 11A, . . ., 21 and the trellis 80 of Figure 8. Furthermore, the circuit 44 left shifts new values into the right sides of Reg0, Reg1, Reg6, and Reg7 as shown in Figures 10B, 11B, . . ., 20B.

[0052]   The recovery circuit 44 operates similarly to the recovery circuit of a conventional Viterbi detector, except that instead of loading the recovered bit sequence — here the sync mark 50 (Figure 7) — into Reg0, Reg1, Reg6, and Reg7, it generates and loads the sync indicator into these registers. Referring to the trellis 80 (Figure 8), for all the state

transitions represented by the normal-line branches 90, the circuit 44 loads 0, 0 into the respective shift registers. Conversely, for the two state transitions represented by the bold-line branches 90, the circuit 44 loads 0, 1 into the respective shift registers. Because the state transitions of the circuit 44 are aligned with the preamble and remain in this alignment during the reading of the sync mark 50, the contents of Reg0, Reg1, Reg6, and Reg7 will eventually converge to the sync.indicator (Figure 7).

[0053]   For purposes of illustration, the sample time t of Figures 10A, 11A, . . ., 21 corresponds to the bits 01 of the preamble (Figure 7) that immediately precede the first inverted group 52 (Figure 7) of the sync mark 50. Consequently, the sample time t+1 corresponds to the last bit 1 of the preamble and the first bit 1 of the group 52. This starting point coincides with the circuit 44 traversing the trellis 80 via the sequence A (Figure 9) of the preamble because the preamble has the state S1 at sample time t, which coincides with the relative sample time k.

[0054]   Also for purposes of illustration, it is assumed that the read signal is noiseless such that read circuit 34 (Figure 3) generates ideal values for the samples B0 and B1. Because the sync mark 50 is known, these ideal samples B0 and B1 are also known. Therefore, the ideal branch metrics are calculated using these ideal samples according to the following known equation:

$$X = (B0_{IDEAL} - B0_{BRANCH})^2 + (B1_{IDEAL} - B1_{BRANCH})^2 \qquad (2)$$

where $B0_{BRANCH}$ and $B1_{BRANCH}$ are the respective sample values 94 and 92 assigned to the branches 90 of Figure 8. The respective ideal path metric for each state S merely equals the sums of the respective ideal branch metrics of the surviving path through the state S at the current sampling time. For clarity, the values of these ideal path metrics label the branches of the respective paths for each state transition in Figures 10A, 11A, . . ., 21.

[0055]   Referring to Figure 10A, the recovery circuit 44 (Figure 6) receives the ideal samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+1. Referring to Figure 7, the two bits represented by the read signal at sample time t+1 are logic 1, logic 1, which respectively correspond to the bit before and the first bit of the inverted group 52. $B0_{IDEAL}$ and $B1_{IDEAL}$ are calculated as follows: $B0_{IDEAL} = 1+1-0-0 = +2$, and $B1_{IDEAL} = 1+1-1-0 = +1$. These calculations are based on the conventional EPR4 sample/bit equations

$$B = A_t + A_{t-1} - A_{t-2} - A_{t-3} \qquad (3)$$

which is disclosed in heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-176 (1678-6)). Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t and the states S at t+1. The updated path metrics label the respective branches. Because the branch metrics X between the states S at sample times t and t+1 are the first branch metrics calculated in this example, the path metrics equal the branch metrics for all branches. For example, the branch metric X, and thus the path metric, for the branch 90a between S0 at t and S1 at S7 equals 4.

[0056]   Next, the recovery circuit 44 identifies the shortest path to each state at sample time t+1, i.e., the surviving paths. For example, referring to state S0 at sample time t+1, there is only one incoming path, so it is the surviving path. Conversely, referring to state S6, there are two incoming paths. Therefore, the recovery circuit 44 selects the incoming branch having the smallest path metric — here the path that includes the branch 90b — as the surviving path. For clarity, the surviving paths are shown in solid line, and the eliminated paths are shown in dashed line. Furthermore, the true path that connects the actual states of the read bits — here the path that includes the branch 90c — has a path metric = 0 in this ideal example.

[0057]   Referring to Figure 10B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7 of the shift register block 46. Reg0, Reg1, Reg6, and Reg7 respectively correspond to the surviving paths ending at the states S0, S1, S6, and S7. For example, referring to Figure 10A, the recovery circuit 44 loads $V0_{t+1} = V1_{t+1} = 0$ into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90d, 90b, and 90c respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 loads $V0_{t+1} = 0$ and $V1_{t+1} = 1$ into Reg1 because the bold-line surviving branch 90a points to the state S1.

[0058]   Referring to Figure 11A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+2. Referring to Figure 7, the next two bits represented by the read signal at sample time t+2 are logic 1 and logic 0, which respectively correspond to the second and third bits of the inverted group 52. Therefore, $B0_{IDEAL} = 1+1-1-1 = 0$, and $B1_{IDEAL} = 0+1-1-1 = -1$. Using $B0_{IDEAL}$ and $B1_{IDEAL}$ the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+1 and the states S at t+2. The updated path metrics label the respective branches. For example, the path metric equals 5 for the path that includes the branch 90i between S6 at t+1 and S0 at t+2.

[0059]   Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+2. For clarity, the sur-

viving paths are shown in solid line, the most recent branches of the newly eliminated paths — here the branches 90f and 90g — are shown in dashed line, and the previously eliminated paths are omitted from the expanded trellis 80. Furthermore, the true path connecting the actual states of the read bits, here the path that includes the branch 90k, has a path metric = 0 in this ideal example.

[0060] Referring to Figure 11B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7 of the shift register block 46. Referring to Figure 11A, the recovery circuit 44 left shifts $V0_{t+2} = V1_{t+2} = 0$ into Reg0, Reg1, and Reg7 because the normal-line surviving branches 90i, 90j, and 90h respectively point to the states S0, S1, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+2} = 0$ and $V1_{t+2} = 1$ into Reg6 because the bold-line surviving branch 90k points to the state S6. Furthermore, the surviving path to S6 now passes through S7 at t+1. Therefore, the circuit 44 loads the path history $V0_{t+1}$ and $V1_{t+1}$ from Reg7 into the corresponding locations of Reg6. Likewise, the surviving path to S7 now passes through S1 at t+1. Therefore, the circuit 44 loads the path history $V0_{t+1}$ and $V1_{t+1}$ from Reg1 into the corresponding locations of Reg7. Moreover, the surviving paths to S0 and S1 now pass through S6 at time t+1. Therefore, the circuit 44 loads the path history $V0_{t+1}$ and $V1_{t+1}$ from Reg6 into the corresponding locations of Reg0 and Reg 1. The circuit 44 performs these register cross loads in a conventional manner such that the prior path histories are cross-loaded and not the updated path histories. For example, if the circuit 44 loaded the path history from Reg6 into Reg1 before it loaded the path history from Reg1 into Reg7, then Reg7 would contain the new path history, not the old path history as is proper, from Reg1.

[0061] Referring to Figure 12A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+3. Referring to Figure 7, the next two bits represented by the read signal at sample time t+3 are logic 0 and logic 0, which respectively correspond to the fourth bit of the inverted group 52 and the first separator bit between the groups 52 and 54. Therefore, $B0_{IDEAL} = 0+0-1-1 = -2$, and $B1_{IDEAL} = 0+0-0-1 = -1$. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+2 and the states S at t+3. The updated path metrics label the respective branches.

[0062] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+3. For clarity, the surviving paths are shown in solid line, the most recent branches of the newly eliminated paths — here the branches 90m and 90n — are shown in dashed line, and the previously eliminated paths are omitted from the expanded trellis 80. Furthermore, the path connecting the actual states of the read bits, here the path that includes the branch 90o, has a path metric = 0 in this ideal example.

[0063] Referring to Figure 12B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 12A, the recovery circuit 44 left shifts $V0_{t+3} = V1_{t+3} = 0$ into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90o, 90p, and 90q respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+3} = 0$ and $V1_{t+3} = 1$ into Reg1 because the bold-line surviving branch 90l points to the state S1. Furthermore, the surviving path to S0 now passes through S6 at time t+2. Therefore, the circuit 44 loads the path history $V1_{t+2} - V0_{t+1}$ from Reg6 into the corresponding locations of Reg0. Likewise, the surviving path to S1 now passes through S0 at t+2, and thus the circuit 44 loads the path history $V1_{t+2} - V0_{t+1}$ from Reg0 into the corresponding locations of Reg1. Moreover, the surviving path to S6 now passes through S7 at t+2, and thus the circuit 44 loads the path history $V1_{t+2} - V0_{t+1}$ from Reg7 into the corresponding locations of Reg6. In addition, because the surviving path the S7 now passes through S7 at t+2, no cross loading of Reg7 is necessary.

[0064] Referring to Figure 13A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+4. Referring to Figure 7, the next two bits represented by the read signal at sample time t+4 are logic 0 and logic 1, which respectively correspond to the second and third separator bits between the groups 52 and 54 of the sync mark 50. Therefore, $B0_{IDEAL} = 0+0-0-0 = 0$, and $B1_{IDEAL} = 1+0-0-0 = +1$. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+3 and the states S at t+4. The updated path metrics label the respective branches.

[0065] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+4. The path connecting the actual states of the read bits, here the path that includes the branch 90s and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0066] Referring to Figure 13B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 13A, the recovery circuit 44 left shifts $V0_{t+4} = V1_{t+4} = 0$ into Reg0, Reg1, and Reg7 because the normal-line surviving branches 90r, 90s, and 90t respectively point to the states S0, S1, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+4} = 0$ and $V1_{t+4} = 1$ into Reg6 because the bold-line surviving branch 90w points to the state S6. Furthermore, the surviving path to S6 now passes through S7 at t+3. Therefore, the circuit 44 loads the path history $V1_{t+3} - V0_{t+1}$ from Reg7 into the respective locations of Reg6. Likewise, the surviving path to S7 now passes through S1 at t+3, and thus the circuit 44 loads the path history $V1_{t+3} - V0_{t+1}$ from Reg1 into the respective locations of Reg7. Moreover, the surviv-

ing path to S1 passes through S0 at t+3, and thus the circuit 44 loads the path history $V1_{t+3}$ - $V0_{t+1}$ from Reg0 into the respective locations of Reg1.

[0067]   Referring to Figure 14A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+5. Referring to Figure 7, the next two bits represented by the read signal at sample time t+5 are logic 1 and logic 0, which respectively correspond to the fourth and fifth separator bits between the inverted group 52 and the group 54 of the sync mark 50. Therefore, $B0_{IDEAL}$ = 1+1-0-0 = +2, and $B1_{IDEAL}$ = 0+1-1-0 = 0. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+4 and the states S at t+5. The updated path metrics label the respective branches.

[0068]   Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+5. The path connecting the actual states of the read bits, here the path that includes the branch 90y and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0069]   Referring to Figure 14B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 14A, the recovery circuit 44 left shifts $V0_{t+5}$ = $V1_{t+5}$ = 0 into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90aa, 90y, and 90z respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+5}$ = 0 and $V1_{t+5}$ = 1 into Reg1 because the bold-line surviving branch 90x points to the state S1. Furthermore, the surviving paths to S6 and S7 now pass through S1 at t+4. Therefore, the circuit 44 loads the path history $V1_{t+4}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg6 and Reg7. Likewise, the surviving path to S1 now passes through S0 at t+4, and thus the circuit 44 loads the path history $V1_{t+4}$ - $V0_{t+1}$ from Reg0 into the respective locations of Reg1. Moreover, the surviving path to S0 now passes through S6 at t+4, and thus the circuit 44 loads the path history $V1_{t+4}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0.

[0070]   Referring to Figure 15A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+6. Referring to Figure 7, the next two bits represented by the read signal at sample time t+6 are logic 0 and logic 0, which respectively correspond to the sixth separator bit between the groups 52 and 54 and the first bit of the group 54 of the sync mark 50. Therefore, $B0_{IDEAL}$ = 0+0-1-1 = -2, and $B1_{IDEAL}$ = 0+0-0-1 = -1. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+5 and the states S at t+6. The updated path metrics label the respective branches.

[0071]   Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+6. The path connecting the actual states of the read bits, here the path that includes the branch 90gg and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0072]   Referring to Figure 15B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 15A, the recovery circuit 44 left shifts $V0_{t+6}$ = $V1_{t+6}$ = 0 into Reg0, Reg1, and Reg7 because the normal-line surviving branches 90gg, 90hh, and 90ff respectively point to the states S0, S1, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+6}$ = 0 and $V1_{t+6}$ = 1 into Reg6 because the bold-line surviving branch 90ii points to the state S6. Furthermore, the surviving path to S6 now passes through S7 at t+5. Therefore, the circuit 44 loads the path history $V1_{t+5}$ - $V0_{t+1}$ from Reg7 into the respective locations of Reg6. Likewise, the surviving path to S7 now passes through S1 at t+5, and thus the circuit 44 loads the path history $V1_{t+5}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg7. Moreover, the surviving paths to S0 and S1 pass through S6 at t+6, and thus the circuit 44 loads the path history $V1_{t+5}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0 and Reg1.

[0073]   Referring to Figure 16A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+7. Referring to Figure 7, the next two bits represented by the read signal at sample time t+7 are logic 0 and logic 1, which respectively are the second and third bits of group 54 of the sync mark 50. Therefore, $B0_{IDEAL}$ = 0+0-0-0 = 0, and $B1_{IDEAL}$ = 1+0-0-0 = 1. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+6 and the states S at t+7. The updated path metrics label the respective branches.

[0074]   Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+7. The path connecting the actual states of the read bits, here the path that includes the branch 90jj and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0075]   Referring to Figure 16B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 16A, the recovery circuit 44 left shifts $V0_{t+5}$ = $V1_{t+5}$ = 0 into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90mm, 90kk, and 90ll respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+7}$ = 0 and $V1_{t+7}$ = 1 into Reg1 because the bold-line surviving branch 90jj points to the state S1. Furthermore, the surviving path to S0 now passes through S6 at t+6, and thus the circuit 44 loads the path history $V1_{t+6}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0. Likewise, the surviving path to S1 now passes through S0 at t+6, and thus

the circuit 44 loads the path history $V1_{t+6}$- $V0_{t+1}$ from Reg0 into the respective locations of Reg1. Moreover, the surviving paths to S6 and S7 now pass through S1 at t+6. Therefore, the circuit 44 loads the path history $V1_{t+6}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg6 and Reg7.

[0076] Referring to Figures 16A and 16B, an analysis of the expanded trellis 80 reveals that the surviving paths have merged into a single path — the path having an ideal path metric equal to 0 — between t and t+3, and $V1_{t+2}$ has merged to a logic 1 in Reg0, Reg1, Reg6, and Reg7. As shown below, $V1_{t+2}$ is the first indicator bit 68 of the sync indicator of Figure 7.

[0077] Referring to Figure 17A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+8. Referring to Figure 7, the next two bits represented by the read signal at sample time t+8 are logic 1 and logic 1, which respectively correspond to the fourth bit of the group 54 and the first separator bit between the groups 55 and 56 of the sync mark 50. Therefore, $B0_{IDEAL}$ = 1+1-0-0 = +2, and $B1_{IDEAL}$ = 1+1-1-0 = +1. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+7 and the states S at t+8. The updated path metrics label the respective branches.

[0078] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+8. The path connecting the actual states of the read bits, here the path that includes the branch 90rr and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0079] Referring to Figure 17B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 17A, the recovery circuit 44 left shifts $V0_{t+8}$ = $V1_{t+8}$ = 0 into Reg0, Reg1, and Reg7 because the normal-line surviving branches 90pp, 90qq, and 90rr respectively point to the states S0, S1, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+8}$ = 0 and $V1_{t+8}$ = 1 into Reg6 because the bold-line surviving branch 90uu points to the state S6. Furthermore, the surviving path to S1 now passes through S0 at t+7. Therefore, the circuit 44 loads the path history $V1_{t+7}$ - $V0_{t+1}$ from Reg0 into the respective locations of Reg1. Likewise, the surviving path to S6 now passes through S7 at t+7, and thus the circuit 44 loads the path history $V1_{t+7}$ - $V0_{t+1}$ from Reg7 into the respective locations of Reg6. Moreover, the surviving path to S7 passes through S1 at t+7, and thus the circuit 44 loads the path history $V1_{t+7}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg7.

[0080] Referring to Figures 17A and 17B, an analysis of the expanded trellis 80 reveals that the surviving paths have merged into a single path — the path having an ideal path metric equal to 0 — between t and t+4.

[0081] Referring to Figure 18A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+9. Referring to Figure 7, the next two bits represented by the read signal at sample time t+9 are logic 1 and logic 0, which respectively are the second and third separator bits between the groups 54 and 56 of the sync mark 50. Therefore, $B0_{IDEAL}$ = 1+1-1-1 = 0, and $B1_{IDEAL}$ =0+1-1-1 = -1. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+8 and the states S at t+9. The updated path metrics label the respective branches.

[0082] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+9. The path connecting the actual states of the read bits, here the path that includes the branch 90zz and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0083] Referring to Figure 18B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 18A, the recovery circuit 44 left shifts $V0_{t+9}$ = $V1_{t+9}$ = 0 into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90yy, 90zz, and 90aaa respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+9}$ = 0 and $V1_{t+9}$ =1 into Reg1 because the bold-line surviving branch 90vv points to the state S1. Furthermore, the surviving path to S0 now passes through S6 at t+8, and thus the circuit 44 loads the path history $V1_{t+8}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0. Likewise, the surviving path to S1 now passes through S0 at t+8, and thus the circuit 44 loads the path history $V1_{t+8}$- $V0_{t+1}$ from Reg0 into the respective locations of Reg1. Moreover, the surviving path to S6 now passes through S7 at t+8, and thus the circuit 44 loads the path history $V1_{t+8}$ - $V0_{t+1}$ from Reg7 into the respective locations of Reg6.

[0084] Referring to Figure 19A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+10. Referring to Figure 7, the next two bits represented by the read signal at sample time t+10 are logic 0 and logic 0, which respectively correspond to the fourth separator bit between the groups 54 and 56 of the sync mark 50 and the first bit of the group 56. Therefore, $B0_{IDEAL}$ = 0+0-1-1 = -2, and $B1_{IDEAL}$ = 0+0-0-1 = -1. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+9 and the states S at t+10. The updated path metrics label the respective branches.

[0085] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+10. The path connecting the actual states of the read bits, here the path that includes the branch 90eee and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0086] Referring to Figure 19B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 19A, the recovery circuit 44 left shifts $V0_{t+10} = V1_{t+10} = 0$ into Reg0, Reg1, and Reg7 because the normal-line surviving branches 90eee, 90fff, and 90ddd respectively point to the states S0, S1, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+10} = 0$ and $V1_{t+10} = 1$ into Reg6 because the bold-line surviving branch 90ggg points to the state S6. Furthermore, the surviving paths to S0 and S1 now pass through S6 at t+9. Therefore, the circuit 44 loads the path history $V1_{t+9}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0 and Reg1. Likewise, the surviving path to S6 now passes through S7 at t+9, and thus the circuit 44 loads the path history $V1_{t+9}$ - $V0_{t+1}$ from Reg7 into the respective locations of Reg6. Moreover, the surviving path to S7 passes through S1 at t+9, and thus the circuit 44 loads the path history $V1_{t+9}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg7.

[0087] Referring to Figure 20A, the recovery circuit 44 receives the next samples $B0_{IDEAL}$ and $B1_{IDEAL}$ corresponding to the sample time t+11. Referring to Figure 7, the next two bits represented by the read signal at sample time t+11 are logic 0 and logic 10, which respectively are the second and third bits of the group 56 of the sync mark 50. Therefore, $B0_{IDEAL} = 0+0-0-0 = 0$, and $B1_{IDEAL} = 1+0-0-0 = +1$. Using $B0_{IDEAL}$ and $B1_{IDEAL}$, the recovery circuit 44 calculates the branch metrics X and updates the path metrics for each of the respective branches 90 between the states S at t+10 and the states S at t+11. The updated path metrics label the respective branches.

[0088] Next, the recovery circuit 44 identifies the surviving path to each state at sample time t+11. The path connecting the actual states of the read bits, here the path that includes the branch 90hhh and into which all the surviving paths will eventually merge, has a path metric = 0 in this ideal example.

[0089] Referring to Figure 20B, once the recovery circuit 44 identifies the surviving paths, it loads the corresponding sync-indicator values V0 and V1 into the respective shift registers Reg0, Reg1, Reg6, and Reg7. Referring to Figure 20A, the recovery circuit 44 left shifts $V0_{t+11} = V1_{t+11} = 0$ into Reg0, Reg6, and Reg7 because the normal-line surviving branches 90kkk, 90iii, and 90jjj respectively point to the states S0, S6, and S7. Conversely, the recovery circuit 44 left shifts $V0_{t+11} = 0$ and $V1_{t+11} = 1$ into Reg1 because the bold-line surviving branch 90hhh points to the state S1. Furthermore, the surviving path to S0 now passes through S6 at t+10, and thus the circuit 44 loads the path history $V1_{t+10}$ - $V0_{t+1}$ from Reg6 into the respective locations of Reg0. Likewise, the surviving path to S1 now passes through S0 at t+10, and thus the circuit 44 loads the path history $V1_{t+10}$ - $V0_{t+1}$ from Reg0 into the respective locations of Reg1. Moreover, the surviving paths to S6 and S7 now pass through S1 at t+10, and thus the circuit 44 loads the path history $V1_{t+10}$ - $V0_{t+1}$ from Reg1 into the respective locations of Reg6 and Reg7.

[0090] Figure 21 is the expanded trellis 80 of Figure 20A with all of the eliminated paths removed. At t+11, the surviving paths have merged into a single surviving path — the path having the ideal path metric equal to 0 — between t and t+8.

[0091] An analysis of the Figures 10A - 21 reveals that the synchronizer circuit 36 of Figure 6 allows the pad 76 to be shorter than the metric needed to accommodate the latency of the recovery circuit 44. Referring to Figures 14B and 15B, the recovery circuit 44 has a latency of ten bits, i.e., five clock cycles. That is, the recovery circuit 44 left shifts ten bits into each of the registers Reg0, Reg1, Reg6, and Reg7 before $V1_{t+2}$ converges to the logic 1 indicator bit 68 in all of these registers. But referring to Figure 14B, $V1_{t+2}$ converges to the logic 1 indicator bit 68 in three of these registers, here Reg1, Reg6, and Reg7, in only eight bits, i.e., four clock cycles. Likewise, referring to Figure 19B, $V1_{t+7}$ converges to the logic 1 indicator bit 70 in three registers Reg0, Reg1, and Reg7 in only four clock cycles. In fact, one finds that the remaining indicator bits 72 and 74 converge to logic 1 in three registers within four clock cycles. The inventors have further discovered that convergence of an indicator bit occurs in Reg1 or Reg6 — depending on whether S1 or S6, respectively, is the current state of the preamble — within four clock cycles. That is, referring to Figures 8 and 9, the state transitions of the sync circuit 36 are aligned with the preamble such that at sample times k the preamble has the state S1 and at times k+1 the preamble has the state S6. Because of this alignment and because the sync mark 50 is merely a partially inverted version of the preamble, the path history stored in Reg1 at times k is accurate beyond the most recent six bits. Likewise, the path history stored in Reg6 at times k+1 is accurate beyond the most recent six bits. Therefore, by analyzing the path history from Reg1 at times k and analyzing the path history from Reg6 at times k+1, one can reduce the length of the pad 76 from eight bits (length required to accommodate the ten-bit latency of the recovery circuit 44) to six bits. A six-bit pad 76 is long enough to allow at least the path history within Reg6 to properly converge to the last indicator bit 74. Thus, this shortening of the pad 76 allows room for two additional data bits in the respective data sector. Because a magnetic disk may have thousands of data sectors, this significantly increases the data-storage capacity of the disk.

[0092] Figure 22 is a block diagram of a disk-drive system 100 according to an embodiment of the invention. Specifically, the disk-drive system 100 includes a disk drive 102, which incorporates the read channel 30 of Figure 3. The disk drive 102 includes a combination write/read head 104, a write-channel circuit 106 for generating and driving the head 104 with a write signal, and a write controller 108 for interfacing the write data to the write-channel circuit 106. In one embodiment, the write-channel circuit 106 includes the data encoder disclosed in heretofore incorporated U.S. Patent Application Ser. No. _____ (Attorney Docket No. 98-S-167 (1678-5)). The disk drive 102 also

includes the read channel 30 for receiving a read signal from the head 104 and for recovering the written data from the read signal, and includes a read controller 114 for organizing the read data. The disk drive 102 further includes a storage medium such as one or more disks 116, each of which may contain data on one or both sides. The write/read head 104 writes/reads the data stored on the disks 116 and is connected to a movable support arm 118. A position system 120 provides a control signal to a voice-coil motor (VCM) 122, which positionally maintains/moves the arm 118 so as to positionally maintain/radially move the head 104 over the desired data on the disks 116. A spindle motor (SPM) 124 and a SPM control circuit 126 respectively rotate the disks 116 and maintain them at the proper rotational speed.

[0093]   The disk-drive system 100 also includes write and read interface adapters 128 and 130 for respectively interfacing the write and read controllers 108 and 114 to a system bus 132, which is specific to the system used. Typical system busses include ISA, PCI, S-Bus, Nu-Bus, etc. The system 100 also typically has other devices, such as a random access memory (RAM) 134 and a central processing unit (CPU) 136 coupled to the bus 132.

[0094]   From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Furthermore, because one of skill in the art can design circuitry or software to perform the above-described functions, the details of this circuitry and software are omitted for clarity.

Claims

1. A synchronizer circuit, comprising:

    an input terminal operable to receive an input signal that includes a synchronization mark;
    an output terminal; and
    a recovery circuit coupled to the input and output terminals and operable to,

        recover the synchronization mark from the input signal, and
        generate a synchronization signal on the output terminal in response to the recovered synchronization mark.

2. The synchronizer circuit of claim 1 wherein the recovery circuit comprises a time-varying Viterbi detector.

3. The synchronizer circuit of claim 1 wherein the recovery circuit is operable to recover the synchronization mark by executing state-transition routines in alignment with the input signal.

4. The synchronizer circuit of claim 1 wherein:

    the input signal includes a preamble; and
    the recovery circuit is operable to,

        recover the synchronization mark from the input signal by sequentially executing respective state-transition routines, and
        determine the alignment between the executing of the state-transition routines and the preamble.

5. The synchronizer circuit of claim 1 wherein:

    the input signal includes a preamble; and
    the recovery circuit is operable to,

        recover the synchronization mark from the input signal by sequentially executing respective state-transition routines, and
        align the executing of the state-transition routines with the preamble.

6. The synchronizer circuit of claim 1, further comprising:

    a control terminal operable to receive a control signal;
    wherein the input signal includes a preamble; and
    wherein the recovery circuit is operable to,

        recover the synchronization mark from the input signal by sequentially executing respective state-transi-

tion routines, and

align the executing of the state-transition routines with the preamble in response to the control signal.

7. The synchronizer circuit of claim 1 wherein:

the input signal includes a preamble; and
the recovery circuit is operable to,

recover the synchronization mark from the input signal by sequentially executing respective state-transition routines,
determine the alignment between the executing of the state-transition routines and the preamble, and
if the alignment is improper, then properly align the executing of the state-transition routines with the preamble.

8. The synchronizer circuit of claim 1, further comprising:

a control terminal operable to receive a control signal; and
wherein response to the control signal, the recovery circuit is operable to generate the synchronization signal in response to a recovered portion of the synchronization mark.

9. The synchronizer circuit of claim 1, further comprising:

a control terminal operable to receive a control signal; and
wherein response to the control signal, the recovery circuit is operable to generate the synchronization signal only if the recovery circuit recovers an aligned portion of the synchronization mark within a predetermined time after the recovery circuit recovers an unaligned portion of the synchronization mark.

10. The synchronizer circuit of claim 1 wherein the recovery circuit is operable to:

recover the synchronization mark by generating a synchronization indicator in response to the synchronization mark; and
generate the synchronization signal in response to the synchronization indicator.

11. The synchronizer circuit of claim 1, further comprising:

a register; and
wherein the recovery circuit is operable to,

recover the synchronization mark by generating a synchronization indicator in response to the synchronization mark;
load the synchronization indicator into the register; and
generate the synchronization signal in response to the loaded synchronization indicator.

12. A circuit, comprising:

an input terminal operable to receive an input signal that includes a first information sequence;
an output terminal; and
a tracking circuit coupled to the input and output terminals and operable to generate a second information sequence from the input signal.

13. A read channel, comprising:

an input terminal operable to receive an input signal that includes a synchronization mark and data;
a synchronization circuit coupled to the input terminal, having a synchronization output terminal, and operable to,

recover the synchronization mark from the read signal, and

generate a synchronization signal on the synchronization output terminal in response to the recovered synchronization mark; and

a Viterbi detector coupled to the input terminal and to the synchronization output terminal and operable to recover the data from the input signal in response to the synchronization signal.

14. The read channel of claim 13 wherein:

the synchronization circuit is operable to transition the synchronization signal from a first to a second signal level; and

the Viterbi detector is operable to begin recovering the data in response to the transition of the synchronization signal.

15. A disk-drive system, comprising:

a data-storage disk having a surface and operable to store data values;
a motor coupled to and operable to rotate the disk;
a read head operable to generate a read signal that includes a data forerunner and data;
a read-head positioning assembly operable to move the read head over the surface of the disk;
a synchronization circuit coupled to the read head and operable to,

recover a synchronization mark from the data forerunner, and

generate a synchronization signal in response to the synchronization mark; and
a Viterbi detector coupled to the read head and to the synchronization circuit and operable to recover the data from the read signal in response to the synchronization signal.

16. A method, comprising:

recovering a synchronization mark from a data signal; and
generating a synchronization signal in response to the recovering of the synchronization mark.

17. The method of claim 16 wherein the recovering comprises tracking a sequence of values represented by the data signal.

18. The method of claim 16 wherein the recovering comprises:

tracking a sequence of values represented by the data signal; and
generating a synchronization indicator in response to the tracked sequence.

19. The method of claim 16 further comprising:

executing data-signal-state transition routines;
aligning the transition routines to the data signal before recovering the synchronization mark; and
wherein the recovering comprises recovering the synchronization mark using the aligned transition routines.

20. The method of claim 16 wherein the recovering comprises executing data-signal-state transition routines in alignment with the data signal.

21. The method of claim 16 further comprising:

executing data-signal-state transition routines; and
determining an alignment between the transition routines and the data signal.

22. The method of claim 16 further comprising:

executing data-signal-state transition routines;
determining an alignment between the transition routines and the data signal; and
realigning the transition routines with the data signal if the determined alignment is undesired.

23. The method of claim 16 wherein:

the recovering comprises recovering a portion of the synchronization mark; and
the generating comprises generating the synchronization signal in response to the recovered portion of the synchronization mark.

24. The method of claim 16 wherein the generating comprises generating the synchronization signal only if the recovery circuit recovers a portion of the synchronization mark within a predetermined time.

25. A method, comprising:

tracking an input signal that includes a first information sequence; and
generating a second information sequence from the input signal.

26. The method of claim 25 wherein the generating comprises generating the second information sequence in response to the first information sequence.

27. The method of claim 25, further comprising loading the second information sequence into a path-history register.

28. A method, comprising:

recovering a synchronization mark from a data signal;
generating a synchronization signal in response to the recovered synchronization mark; and
recovering data from the data signal in response to the synchronization signal.

29. The method of claim 28 wherein:

the generating comprises,

generating the synchronization signal having a first signal level, and
transitioning the synchronization signal from the first to a second signal level in response to recovering the synchronization mark; and

the recovering the data comprises initiating recovery of the data in response to the transitioning of the synchronization signal.
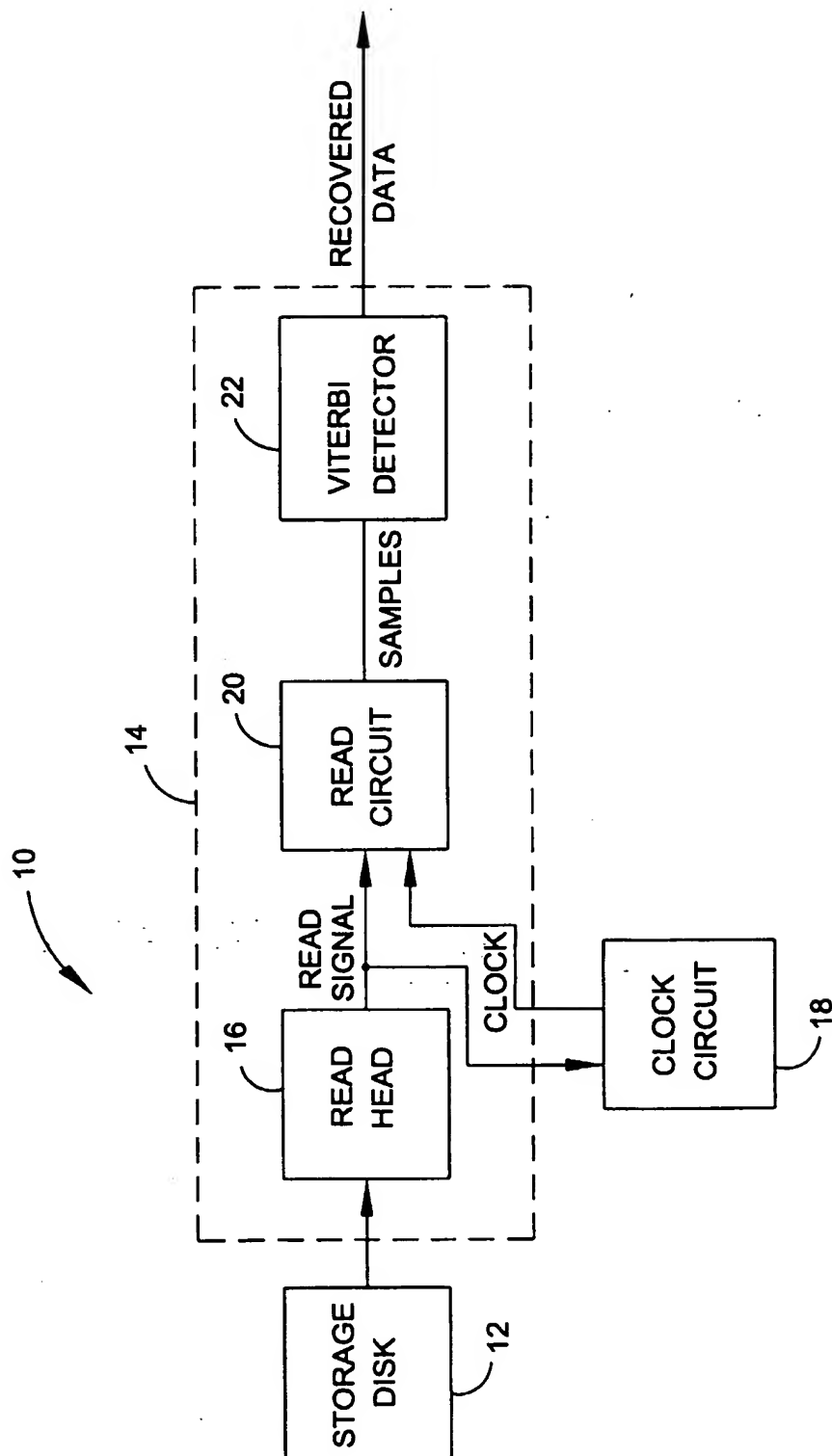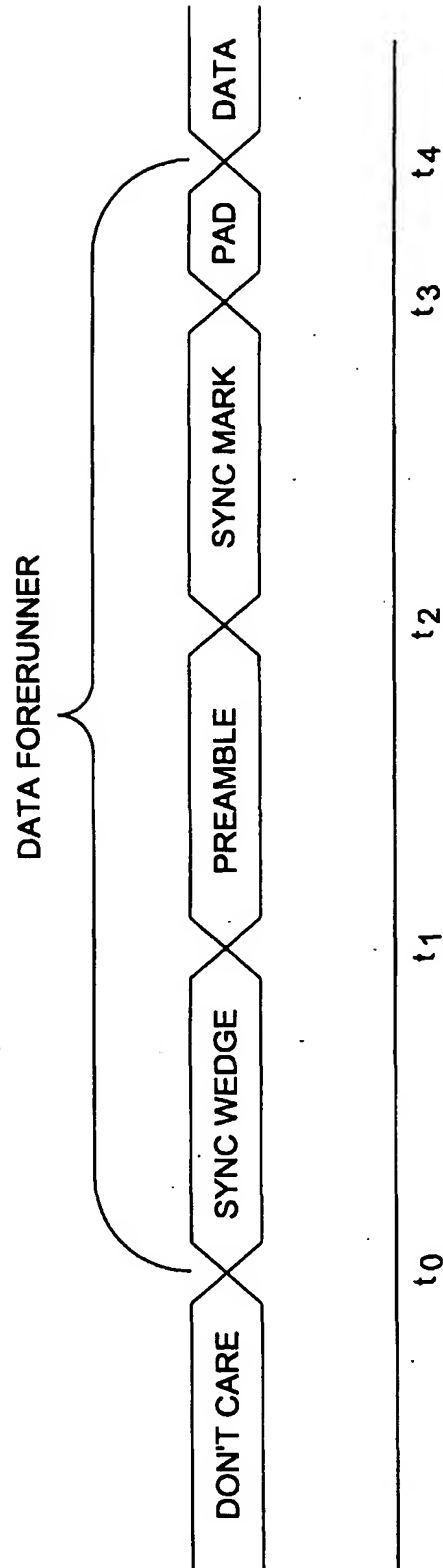
Fig.1 (Prior Art)

DATA FORERUNNER

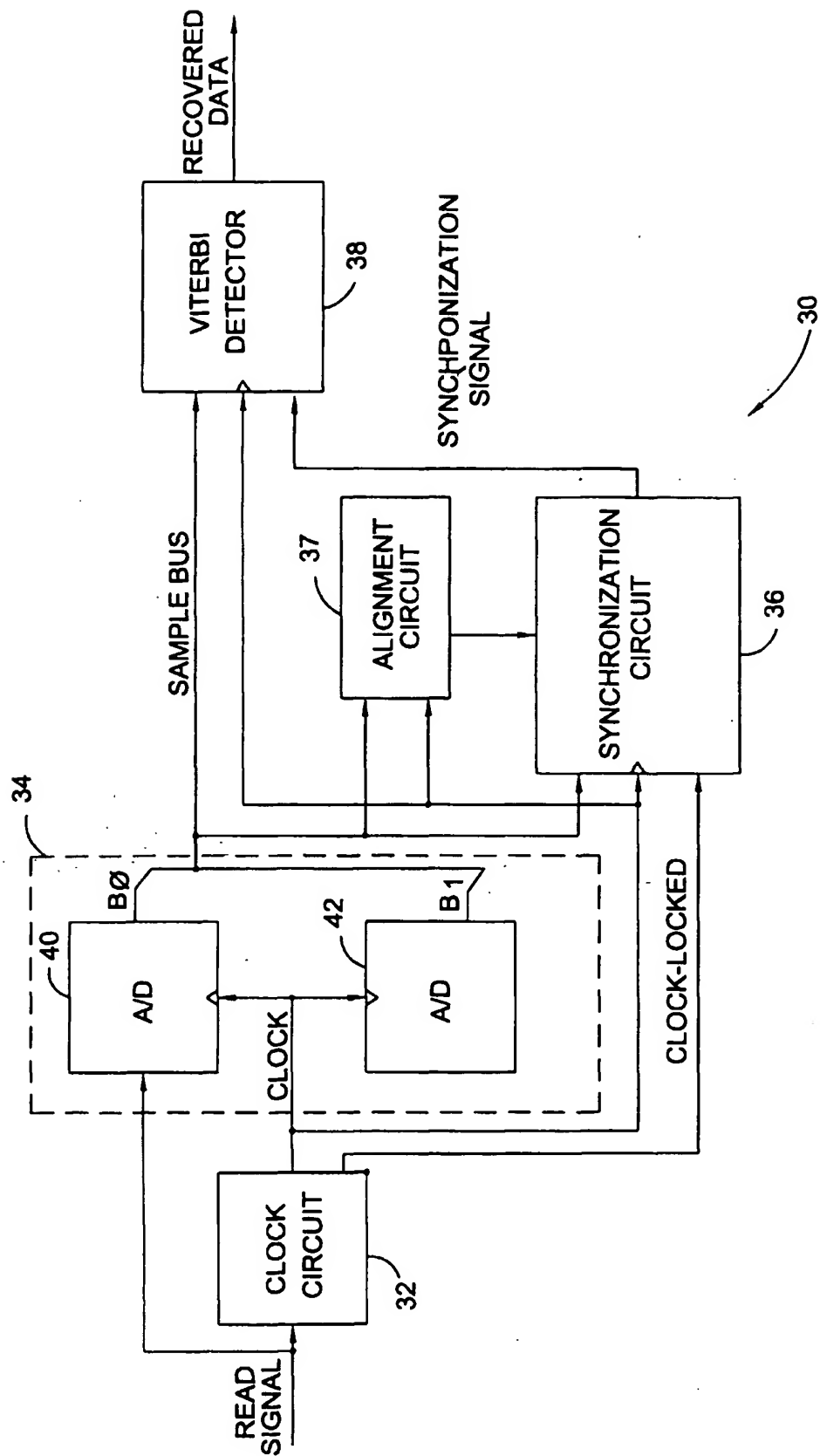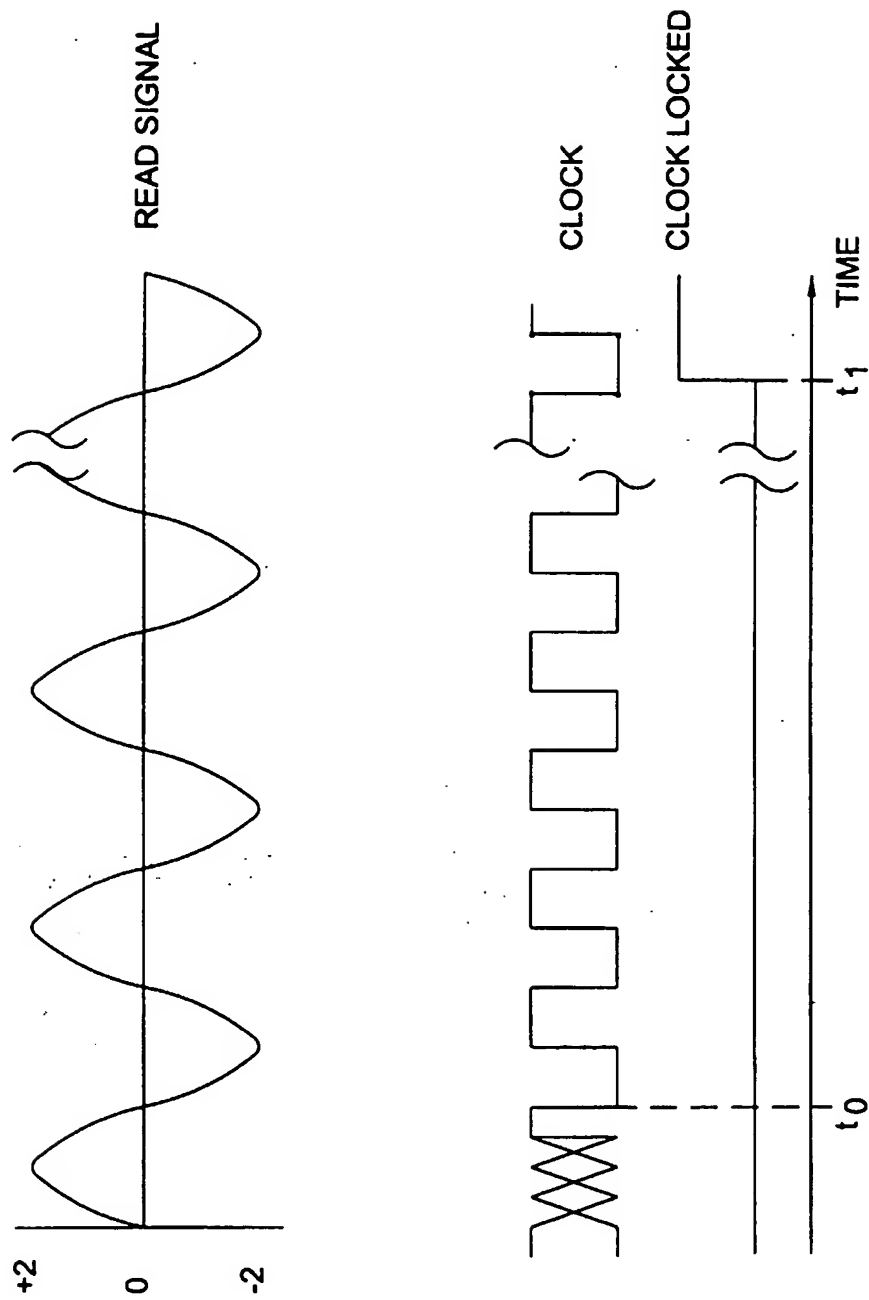| DON'T CARE	SYNC WEDGE	PREAMBLE	SYNC MARK	PAD	DATA

t_0 t_1 t_2 t_3 t_4

Fig.2 (Prior Art)

RECOVERED
DATA

VITERBI
DETECTOR

38

SYNCHPONIZATION
SIGNAL

30

SAMPLE BUS

ALIGNMENT
CIRCUIT

37

SYNCHRONIZATION
CIRCUIT

36

34

40

A/D

B∅

42

A/D

B 1

CLOCK

CLOCK-LOCKED

CLOCK
CIRCUIT

32

READ
SIGNAL

Fig.3

Fig.4

Fig.5

READ SIGNAL

CLOCK SIGNAL

SYNCHRONIZATION SIGNAL

TIME

t_2 t_3 t_4

SYNCHRONIZER CIRCUIT

RECOVERY CIRCUIT

44

SHIFT REGISTERS

46

36

SYNCHRONIZATION SIGNAL

TO VITERBI DETECTOR 38(FIG. 3)

CLOCK

from CLOCK CIRCUIT 32 (FIG. 3)

SAMPLE BUS

from READ CIRCUIT 34 (FIG. 3)

ALIGNMENT SELECT

CLOCK LOCKED

SYNC-MARK

INDICATOR SELECT

TIME-OUT SELECT

Fig.6

PREAMBLE 0011001100110011001100110011001100

SYNC-MARK AND PAD 0011110000110000011111100000011100

PR ⊕ SM 000011110000000111000011110011110000000

SYNC-INDICATOR 000000010000000000000000010000001000000

time ⟶

Fig.7

Fig.8

K

A −2,0 001 +2,0 K+1 110 −2,0 K 001 +2,0 K+1 110 −2,0 K 001 —

B 0,+2 011 0,−2 100 0,+2 011 0,−2 100 0,+2 011 —

C 0,−2 100 0,+2 011 0,−2 100 0,+2 011 0,−2 100 —

D +2,0 110 −2,0 001 +2,0 110 −2,0 001 +2,0 110 —

I

Fig.9

Fig.10A

Fig.10B

Fig.11A

Fig.11B

Fig.12A

V0₍ₜ₊₁₎ V1₍ₜ₊₁₎ V0₍ₜ₊₂₎ V1₍ₜ₊₂₎ V0₍ₜ₊₃₎ V1₍ₜ₊₃₎

Reg 0: 0 0 0 1 0 0

Reg 1: 0 0 0 0 0 1

Reg 6: 0 1 0 0 0 0

Reg 7: 0 1 0 0 0 0

Fig.12B

Fig.13A

$V0_{t+1}$ $V1_{t+1}$ $V0_{t+2}$ $V1_{t+2}$ $V0_{t+3}$ $V1_{t+3}$ $V0_{t+4}$ $V1_{t+4}$

Reg 0: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Reg 1: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Reg 6: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Reg 7: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Fig.13B

Fig.14A

$V0_{t+1}$ $V1_{t+1}$ $V0_{t+2}$ $V1_{t+2}$ $V0_{t+3}$ $V1_{t+3}$ $V0_{t+4}$ $V1_{t+4}$ $V0_{t+5}$ $V1_{t+5}$

Reg 0: 0 1 0 0 0 0 1 0 0

Reg 1: 0 0 0 1 1 0 0 1 0 1

Reg 6: 0 0 0 1 1 0 0 0 0 0

Reg 7: 0 0 0 1 1 0 0 0 0 0

Fig.14B

Fig.15A

Reg 0

$V0_{t+1}$	$V1_{t+1}$	$V0_{t+2}$	$V1_{t+2}$	$V0_{t+3}$	$V1_{t+3}$	$V0_{t+4}$	$V1_{t+4}$	$V0_{t+5}$	$V1_{t+5}$	$V0_{t+6}$	$V1_{t+6}$
0	0	0	1	0	0	0	0	0	0	0	0

Reg 1

| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Reg 6

| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Reg 7

| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Fig.15B

Fig.16A

		$V0_{t+1}$	$V1_{t+1}$	$V0_{t+2}$	$V1_{t+2}$	$V0_{t+3}$	$V1_{t+3}$	$V0_{t+4}$	$V1_{t+4}$	$V0_{t+5}$	$V1_{t+5}$	$V0_{t+6}$	$V1_{t+6}$	$V0_{t+7}$	$V1_{t+7}$
Reg 0				0	0	0	1	0	0	0	0	0	0	1	0	0
Reg 1				0	0	1	0	0	0	0	0	0	0	0	0	1
Reg 6				0	0	1	0	0	0	0	0	0	0	0	0	0
Reg 7				0	0	1	0	0	0	0	0	0	0	0	0	0

Fig.16B

Fig.17A

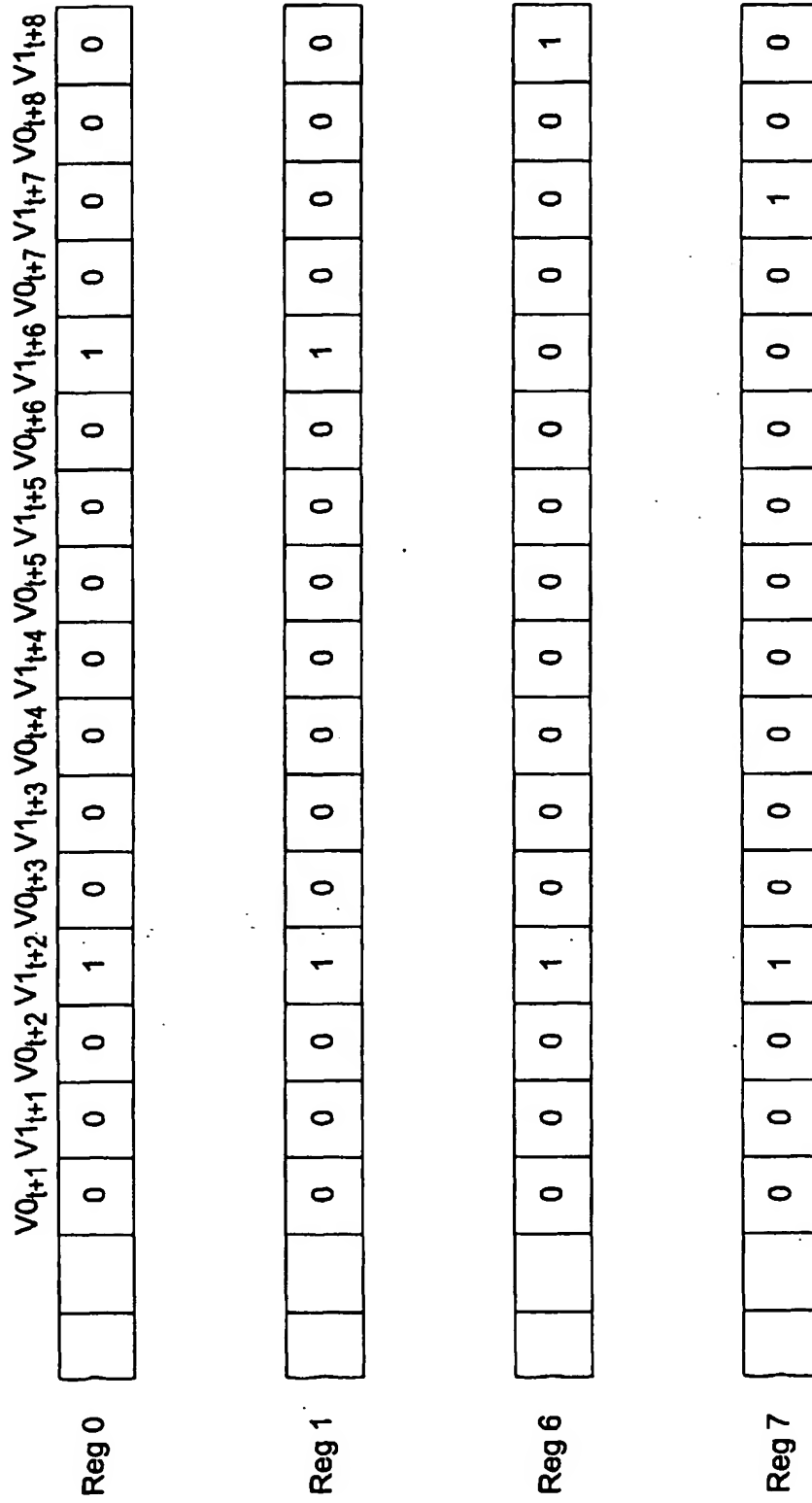|	$V0_{t+1}$	$V1_{t+1}$	$V0_{t+2}$	$V1_{t+2}$	$V0_{t+3}$	$V1_{t+3}$	$V0_{t+4}$	$V1_{t+4}$	$V0_{t+5}$	$V1_{t+5}$	$V0_{t+6}$	$V1_{t+6}$	$V0_{t+7}$	$V1_{t+7}$	$V0_{t+8}$	$V1_{t+8}$
Reg 0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Reg 1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Reg 6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Reg 7	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Fig. 17B

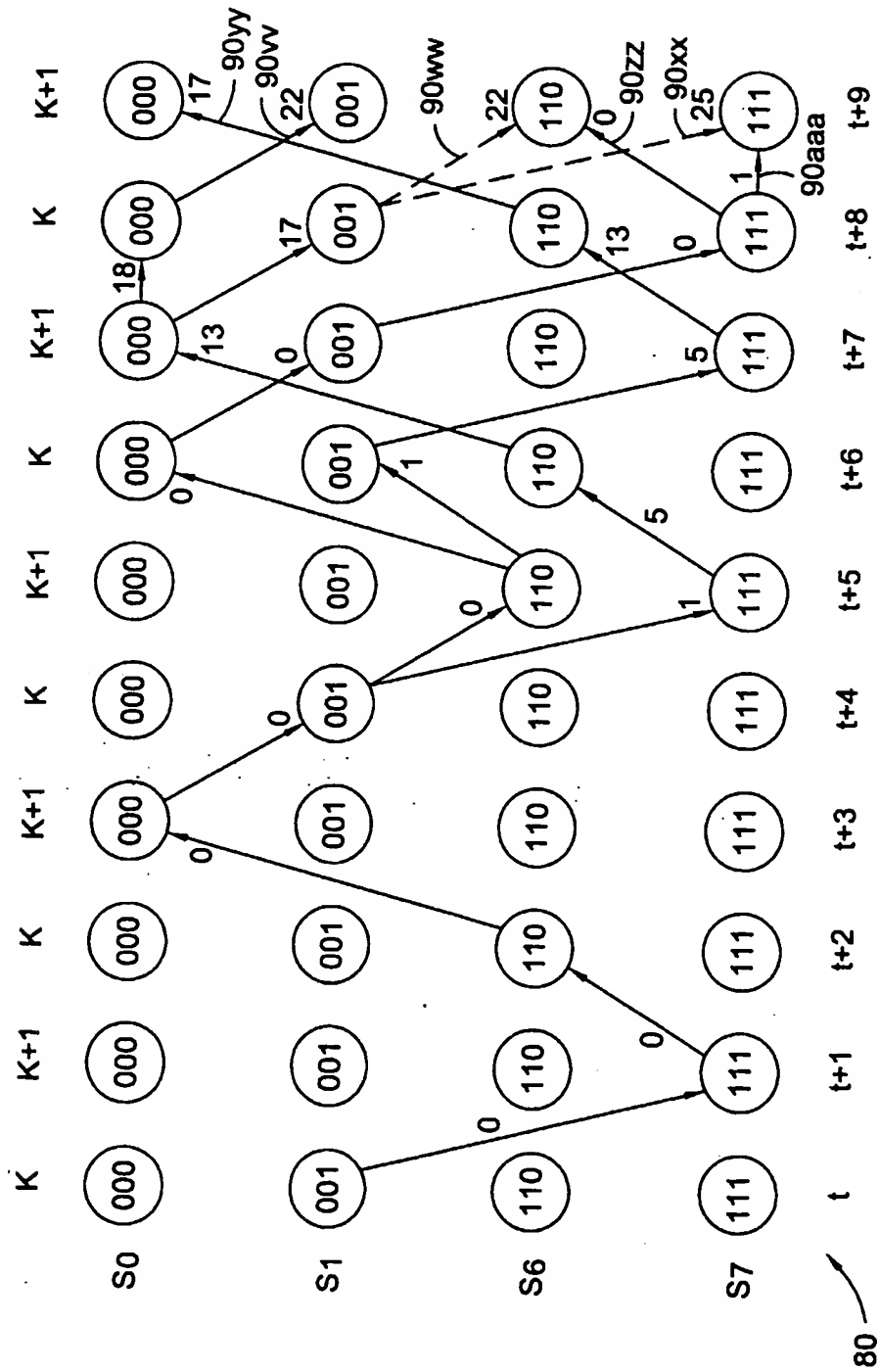*Fig.18A*

$V0_{t+1}$ $V1_{t+1}$ $V0_{t+2}$ $V1_{t+2}$ $V0_{t+3}$ $V1_{t+3}$ $V0_{t+4}$ $V1_{t+4}$ $V0_{t+5}$ $V1_{t+5}$ $V0_{t+6}$ $V1_{t+6}$ $V0_{t+7}$ $V1_{t+7}$ $V0_{t+8}$ $V1_{t+8}$ $V0_{t+9}$ $V1_{t+9}$

Reg 0 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0

Reg 1 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1

Reg 6 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0

Reg 7 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0

Fig.18B

Fig.19A

VO_{t+1} $V1_{t+1}$ VO_{t+2} $V1_{t+2}$ VO_{t+3} $V1_{t+3}$ VO_{t+4} $V1_{t+4}$ VO_{t+5} $V1_{t+5}$ VO_{t+6} $V1_{t+6}$ VO_{t+7} $V1_{t+7}$ VO_{t+8} $V1_{t+8}$ VO_{t+9} $V1_{t+9}$ VO_{t+10} $V1_{t+10}$

Reg 0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Reg 1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Reg 6	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
Reg 7	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

Fig.19B

47

Fig.20A

VO$_{t+2}$ V1$_{t+2}$ VO$_{t+3}$ V1$_{t+3}$ VO$_{t+4}$ V1$_{t+4}$ VO$_{t+5}$ V1$_{t+5}$ VO$_{t+6}$ V1$_{t+6}$ VO$_{t+7}$ V1$_{t+7}$ VO$_{t+8}$ V1$_{t+8}$ VO$_{t+9}$ V1$_{t+9}$ VO$_{t+10}$ V1$_{t+10}$ VO$_{t+11}$ V1$_{t+11}$

Reg 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0

Reg 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1

Reg 6 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
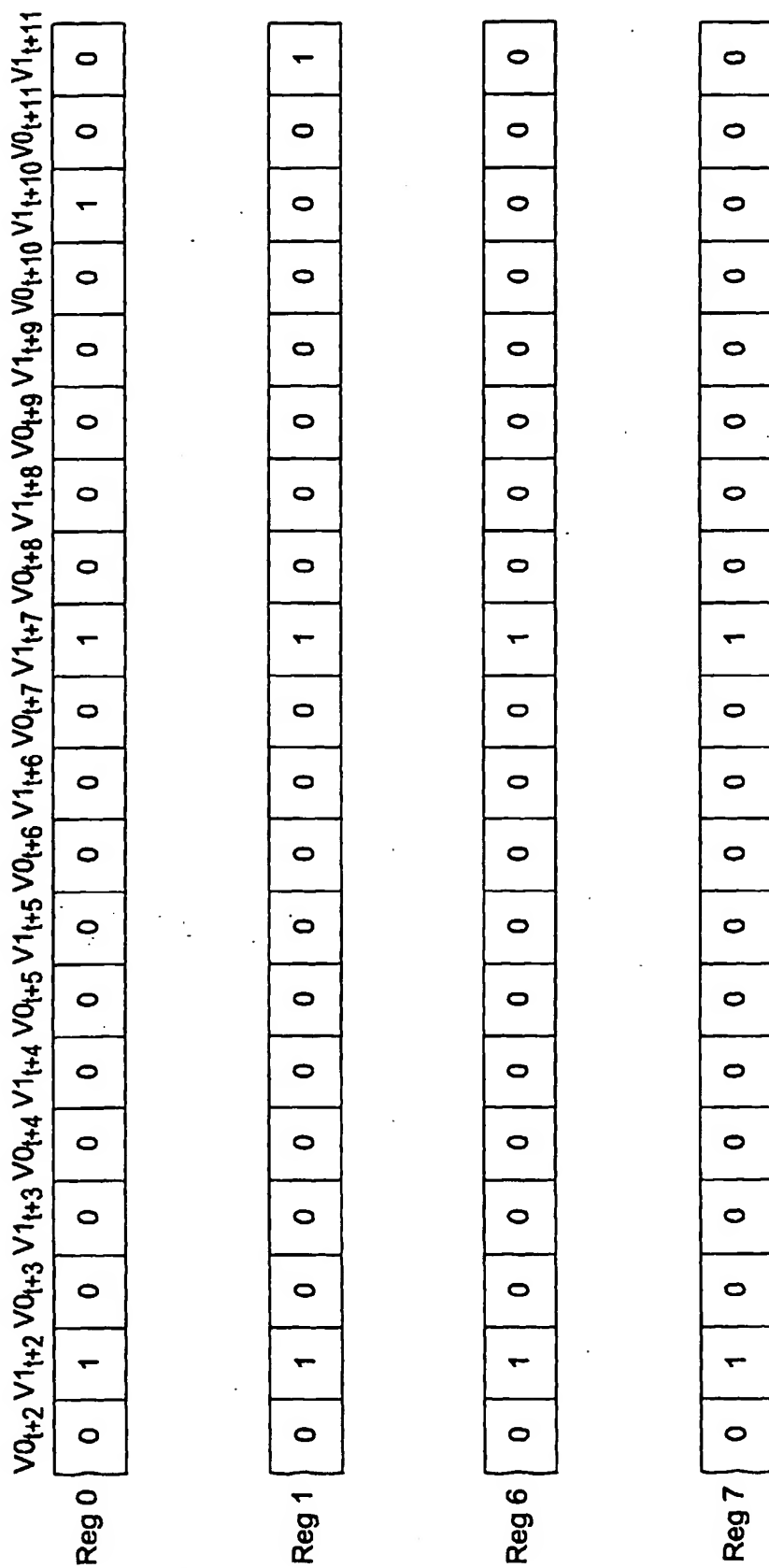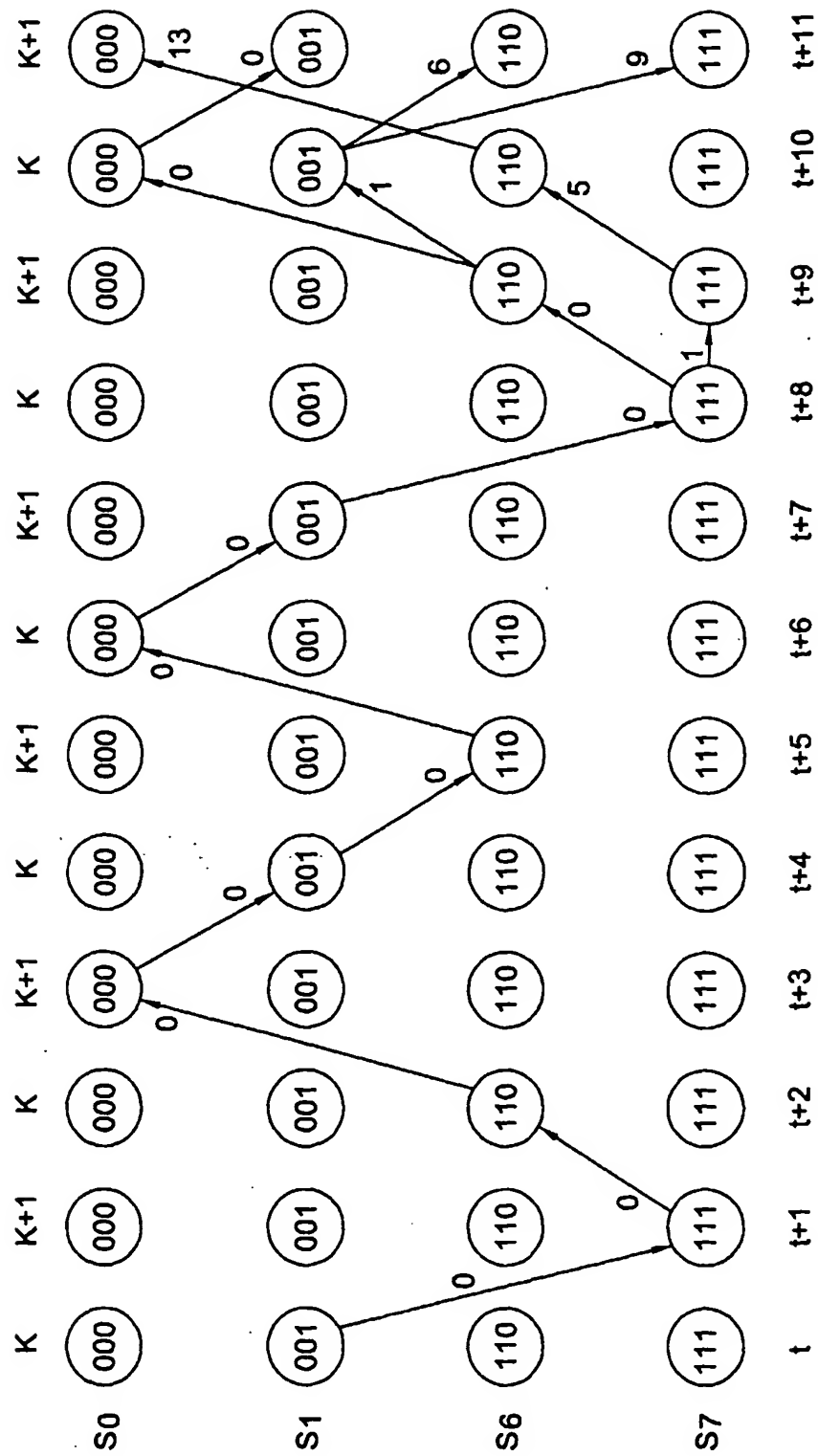
Reg 7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0

Fig.20B

Fig.21

Fig.22